INSTRUCTION MANUAL

PakBus Networking Guide for the CR10X, CR510, CR23X, and CR200 Series and LoggerNet 2.1C

Revision: 3/05

PakBus Networking Table of Contents

PDF viewers note: These page numbers refer to the printed version of this document. Use the Adobe Acrobat® bookmarks tab for links to specific sections.

Appendices

Figures

PakBus Networking Guide

1. Introduction

We know the advantages of email vs. a telephone conversation:

- A real time connection is not necessary as the network accepts messages at any time.
- Network devices have buffers to handle multiple messages on the network.
- The network itself intelligently optimizes routes.
- Backup routes are possible.

In similar fashion, PakBus improves upon traditional connection-based datalogger communications. PakBus devices such as LoggerNet and dataloggers with a PakBus operating system communicate by putting packets of data onto the network. Packets have destination and source addresses and are size limited to, typically, 1000 bytes. In the Quick Start example network, to collect data from a CR205, LoggerNet sends a Collect Data packet to the CR10XPB router which passes it on to the CR205. The CR205 sends back Collect Data Response packets until data collection is complete.

PakBus gives dataloggers the ability to *concurrently communicate* with multiple peripherals at the same time.

PakBus networks have the distributed routing intelligence to continually evaluate links, optimizing delivery times and, in case of device failure, allowing automatic switchover to a configured backup route.

Guide Summary

Quick Start, Section 2, introduces PakBus and steps through the setup of a small PakBus network.

Choosing a Network Configuration, Section 3, summarizes PakBus basics and provides information toward choosing the best approach to your application network.

Sections 4, 5, and 6 show how to set up PakBus devices (LoggerNet, dataloggers, and communications peripherals), and gives descriptions of PakBus instructions for the CR10X, CR510 and CR23X.

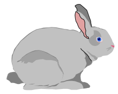
PakBus Concepts, Section 7, provides more detail regarding PakBus devices and fundamentals.

Troubleshooting, Section 8, helps to identify problems during the learning process.

Appendix A is a Glossary of PakBus networking terms used in this guide.

Appendices B to H present examples of various types of PakBus networks.

2. Quick Start – A Basic PakBus Network



Ingredients of a PakBus Network

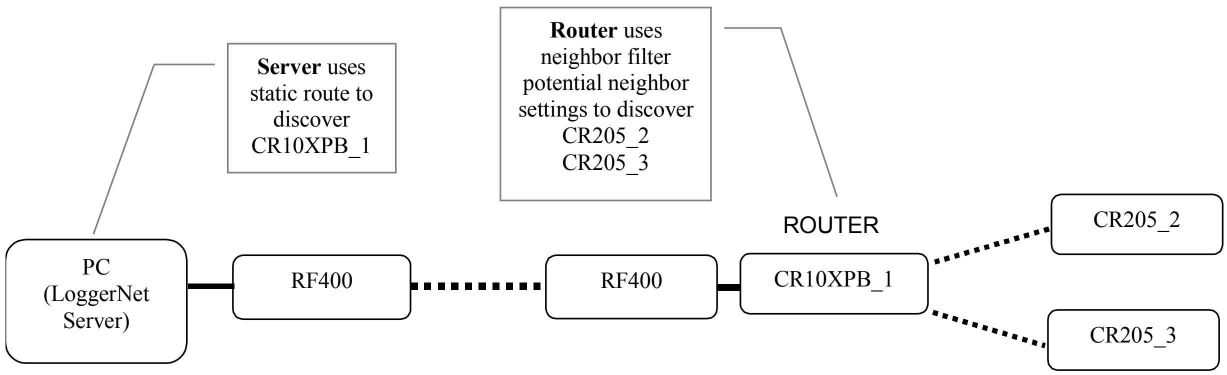
- LoggerNet (server).
- PakBus devices (dataloggers with PakBus OS or native PakBus devices such as the CR200 Series).
- Routers (if used; typically dataloggers with PakBus OS and configured as routers).
- Communication peripherals (RF400 Series, COM210, COM310, NL100, etc.).

Setup of a PakBus Network

- Create the device map in LoggerNet.
- Assign each PakBus device a unique address from 1 to 3999 (LoggerNet defaults to address 4094).
- Create datalogger-routers by entering *D15 settings for: max network nodes, max device neighbors, max network routers.
- Configure routers to discover by neighbor filter or by beaconing the next device (neighbor) toward the destination.

Router configuration can be done with keyboard display using *D15 or by sending a *.dld program with network settings having been configured in Edlog \ Options \ PakBus Settings.

- Set up communication peripheral interfaces to match the attached datalogger's beacon or neighbor filter port.
- Set up all network RF400 Series radios with the same: hopping sequence, network address, radio address and, normally, with standby mode of "< 24 mA always on" and Active Interface of "CSDC 7" (AutoSense if connecting to a PC).



For this Quick Start network example you will need the following items or the equivalent:

1. Compatible PC with LoggerNet 2.1c or higher and PakCom software.
2. CR10XPB or CR510PB datalogger (with PakBus OS).
3. CR10KD keyboard display.
4. 2 CR205 (or CR210, CR215) dataloggers with antennas.
5. 2 RF400 (or RF410, RF415) Radios with antennas and antenna cables.
6. CSI # 15966 12 VDC wall adapter power supply for base RF400.
7. PS100 or equivalent 12V power supply for CR10XPB.
8. 2 CSI # 16869 sealed rechargeable batteries for CR205s.
9. Two 109-L temperature probes or two 22KΩ/10KΩ resistor sets to simulate 109-Ls.
10. SC32B RS-232 to CS I/O interface.

Step 1 – Set Up Base RF400 Series Radio

a. Select an RF400 and label it “base”

b. Restore default settings as follows:

- Connect a serial cable from your PC’s COM port to the base RF400’s RS-232 port.
- Run PakCom and configure it to your PC’s COM port (1 – 4).
- Connect RF400 to its 12 VDC power supply.
- Click on PakCom Radio Settings / OS Download tab and click on Settings / OS.

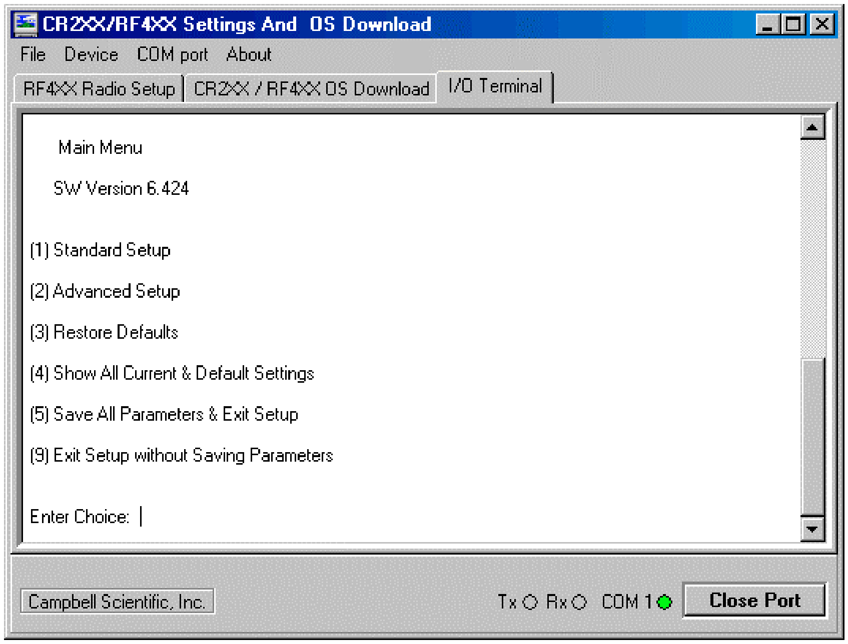


FIGURE 1. PakCom I/O Terminal Screen

- Click on I/O Terminal, click on Open Port, and press the RF400's "Program" green button.

- Press "3" to Restore Defaults.

  You have just selected (among other things) the default Hopping Sequence of "0". It is best to have an 'unused' Hopping Sequence. You can determine if other RF400s near your location are using a Hopping Sequence by powering down all but one RF400 (with antenna) and watching for activity in the green LED over a period of time (close LoggerNet for this test). Other available Hopping Sequences are 1 to 6. If you select another Hopping Sequence, set all network radios to the same setting.

- Configure Standby Mode to "< 24 mA always on"

- Leave other settings at defaults (including the Net Address and Radio Address and leave the Active Interface in "AutoSense" mode).

- Save All Parameters and Exit Setup.

Step 2 – Set up Remote RF400 Series

a. Label the other RF400 "remote".

b. Restore Default settings.

c. Verify that remote RF400 is set up for the same Hopping Sequence, Net Address, and Radio Address as the base RF400.

d. Configure Standby Mode to "< 24 mA always on"

e. In Standard Setup configure the Active Interface = Datalogger CSDC (by default this will be CSDC 7).

f. Save All Parameters and Exit Setup.

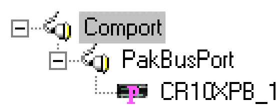
Step 3 – Establish Temporary Direct Communications to CR10XPB_1

a. Connect CR10XPB to 12 VDC power supply and turn on 12V power.

b. Using an SC12 cable and SC32B, connect the CR10XPB to the LoggerNet PC's active COM port.

c. Attach CR10KD keyboard display to CR10XPB using SC12 cable.

d. In LoggerNet Setup, configure this device map (and apply it).



(leave all settings defaults)

e. In the LoggerNet Setup device map make the PakBus Address for CR10XPB_1 = 1.

f. Set the CR10XPB's PakBus Address = 1 as follows:

- Press *D15 on keyboard/display.
- Key "0001" into window 15: and press *0 to save the new setting.

Disregard for now the other windows of *D15.

Step 4 – Write CR10XPB Program

a. Open Edlog in LoggerNet, start a new CR10X-TD program, key in the following (TD programs are used in TD and PB dataloggers). This program will serve as a vehicle for carrying the PakBus Address and router settings to CR10XPB_1. The program also illustrates that, although functioning as a router, the datalogger can still perform all standard datalogger functions such as taking measurements and storing data.

```
*Table 1 Program
  01:  1                Execution Interval (seconds)

1:  Batt Voltage (P10)
  1:    1               Loc [ Battery      ]

2:  Internal Temperature (P17)
  1:    2               Loc [ Panel_Temp     ]
```

```
3:  Data Table (P84)
   1:    0                 Seconds into Interval
   2:    10                Seconds Interval
   3:    0                 (0 = auto allocate, -x = redirect to inloc x)
   4:    BASIC             Table Name

4:  Sample (P70)
   1:    1                 Reps
   2:    1                 Loc [ Battery       ]

5:  Sample (P70)
   1:    1                  Reps
   2:    2                 Loc [ Panel_Temp       ]
```

b. Save as "BASIC.CSI" in Edlog.

Step 5 – Do CR10XPB PakBus Setup in Edlog and Send it to Datalogger

a. Click on Edlog, Options, PakBus Settings.

b. Configure as shown below

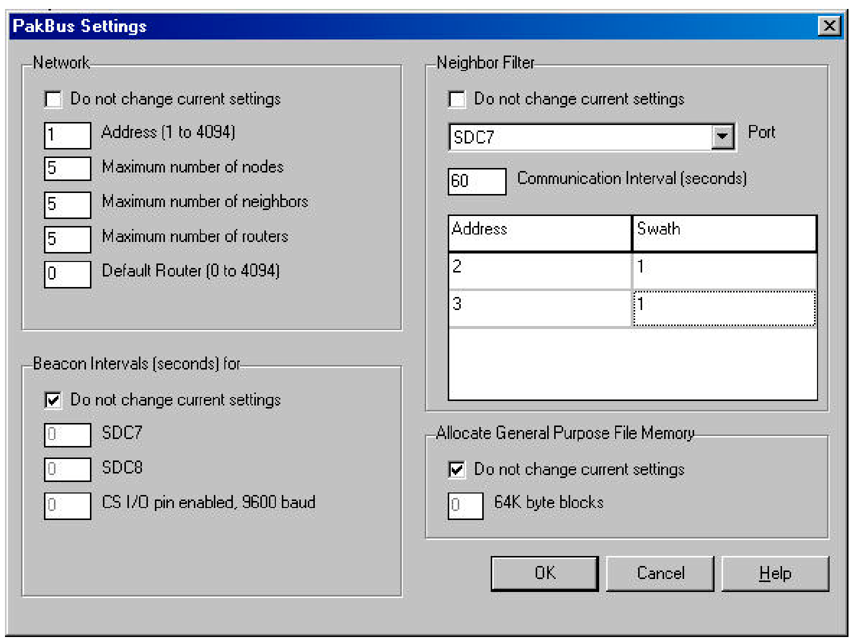


FIGURE 2. Edlog PakBus Settings

When sent to the datalogger, the program with the above settings:

- Sets the CR10XPB PakBus Address to 1.
- Configures the CR10XPB as a router by entering non-zero values in Max Nodes, Max Neighbors, and Max Routers.
- Sets up a Neighbor Filter in the CR10XPB program which can discover listed PakBus devices using the datalogger's CSDC 7 port to RF400; sets the Communications Verification Interval for 60 seconds; lists as potential neighbors the devices with addresses 2 and 3 (future CR200 Series PakBus Addresses).

CAUTION *D15 settings allocate memory similar to *A settings. Changing *D15 settings later on could result in loss of data.

It is a good idea, when configuring *D15 settings, to leave 'room to grow,' however, the datalogger routing tables use memory, so avoid using overly large numbers.

a. Click OK to write these settings to the program.

b. Save and Compile BASIC.CSI.

c. Connect in LoggerNet to the CR10XPB and send BASIC.dld to the datalogger.

Step 6 – Write CR205 Program

a. Verify that the latest OS (example, cr2osv02_20.a43) has been downloaded to the CR205s. This can be done by connecting to a CR205 in PakCom, clicking on Monitor Data Tables, entering "1" in Update in Seconds field, and observing the Data Table called "Status" which shows the CR205's current OS Version.

b. Verify that the program compiler matches the OS is installed in your CSIEdit (example, cr2compv02_20.exe).

c. Open CRBasic in LoggerNet, start a new CR200 program file, and key in the following code.

```
'CR200 Datalogger

Public AirTemp_C

DataTable (AirTemp,1,1000)
  DataInterval (0,1,min)
  Average (1,AirTemp_C,0)
EndTable
```

```
BeginProg
Scan(10,sec)
'Make Measurements
Therm109 (AirTemp_C,1,1,Ex1,1.0,0)
CallTable AirTemp
NextScan
EndProg
```

d. Save and compile the program as “TEMP_PROBE.CR2”.

Step 7 – Set up CR205s and Download Program

a. Label one CR205 as “CR205_2” and the other as “CR205_3” representing their future PakBus Addresses.

b. Connect the CR205s to 12VDC supplies.

c. Using an SC12 cable, connect CR205_2’s RS-232 port to an available COM port. Make sure that no other software is running that uses this COM port (including LoggerNet).

d. Run PakCom software.

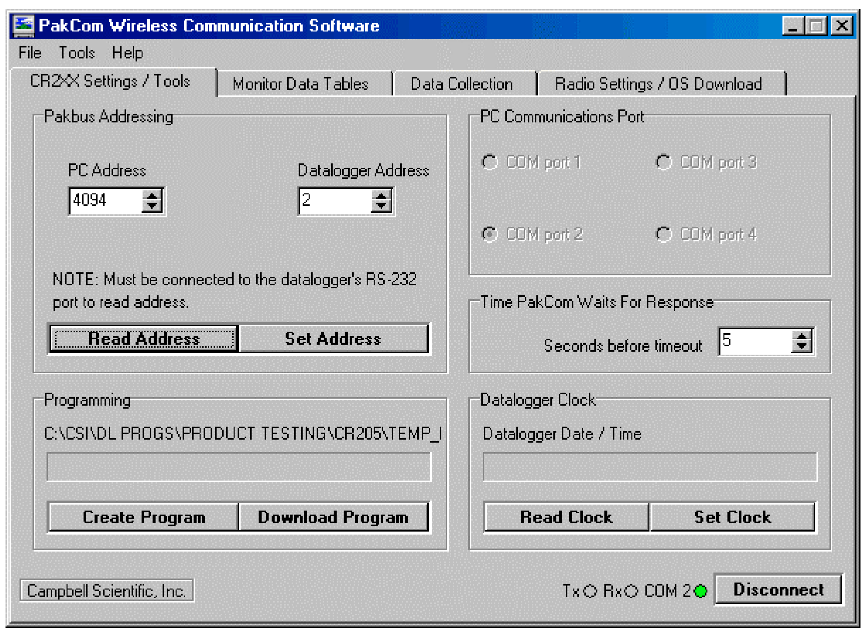
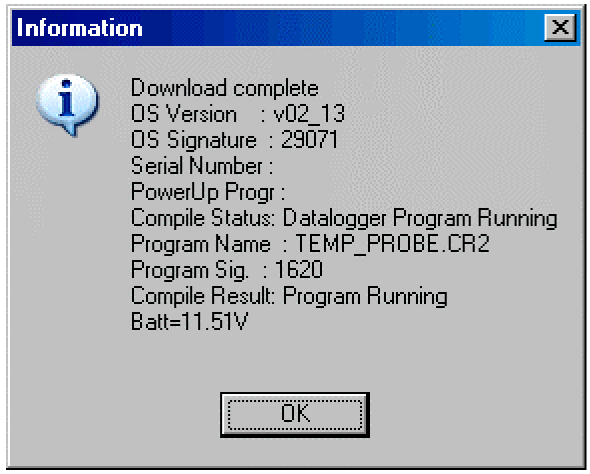


FIGURE 3. PakCom CR2XX Settings / Tools

e. Select the available COM port in CR2XX Settings/Tools, press Connect, and then press Read Address. The CR205’s current PakBus Address (“Datalogger Address”) will be shown.

f. Enter “2” in the Datalogger Address field and click Set Address button to make CR205_2’s Datalogger Address = 2.

g. Using Download Program button, download TEMP_PROBE.BIN to the datalogger. Check the Information window to be sure the program is running.

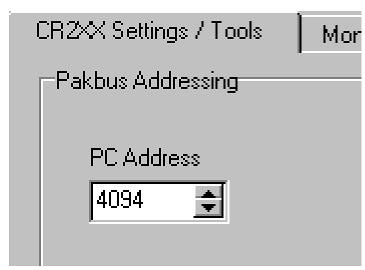


h. Repeat Step 7 with CR205_3 except make its PakBus Address (Datalogger Address) = 3.

Step 8 – Set up CR205 Radios

Do the following setup for CR205_2 and CR205_3 in turn.

a. Using PakCom, verify that the PC Address is 4094 (LoggerNet's PakBus Address).



b. Click on the PakCom tab entitled "Radio Settings / OS Download" and click on the Read Settings button.

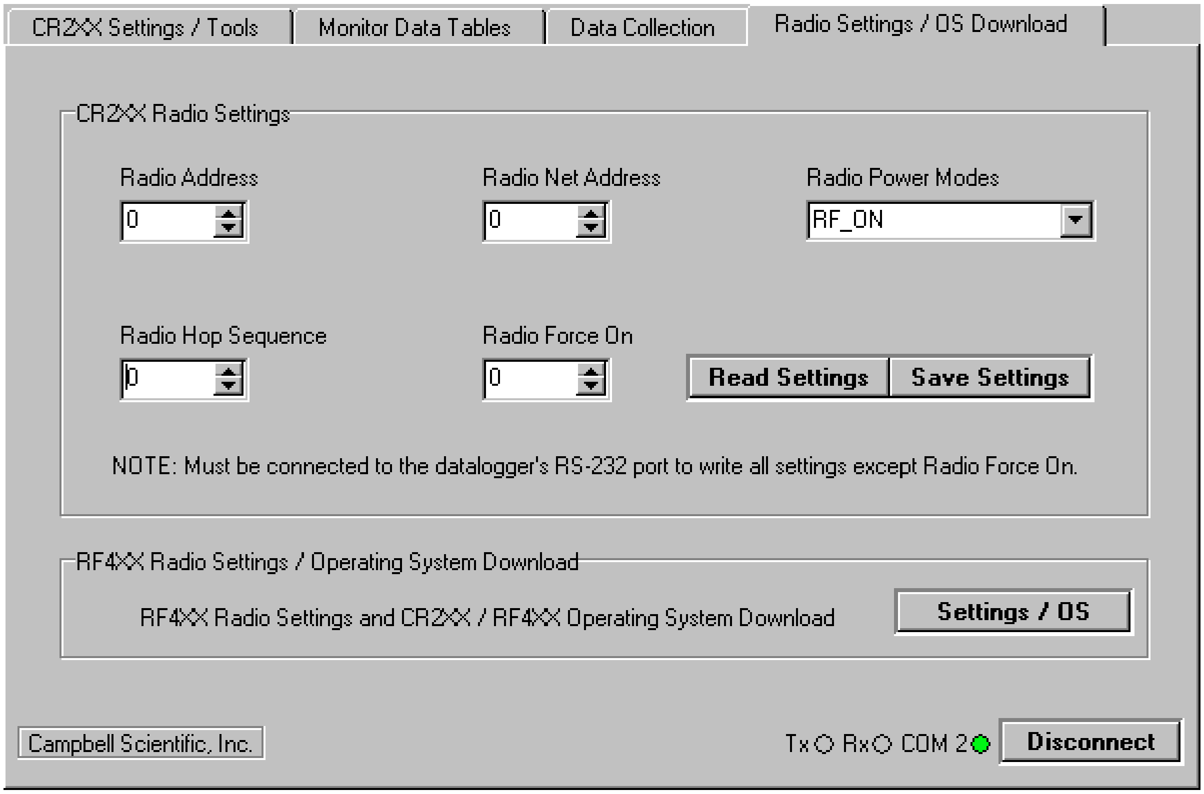


FIGURE 4. PakCom Radio Settings / OS Download

c. Select the Radio Hop Sequence decided upon for the RF400s in Step 2.

d. Select the Radio Net Address decided upon in Step 2.

e. Leave Radio Address = 0.

f. Select Radio Power Mode = RF_ON (radio not duty cycling).

g. Click Save Settings button to transfer settings to CR205.

h. Repeat Step 8 for an exact match in these settings with CR205_3.

i. At this point, the only difference between CR205_2 and CR205_3 is their PakBus Addresses. Use PakCom to connect and check their respective status tables to be sure.

Step 9 – Wire 109-L Temperature Sensors or Equivalent

a. Wire the 109-L temperature sensors to SE1, ⏚, and EX2 on the CR205s per probe manual for both CR205s.

or

b. If you prefer to use fixed resistors to simulate temperature probes, obtain 10KΩ and 22KΩ, 5%, ¼ Watt resistors (tolerance and wattage not

critical). On each CR205 connect a 10KΩ resistor from EX1 to SE1 and a 22KΩ resistor from SE1 to the adjacent ⏚ (signal ground). These resistors simulate a 109 thermister probe at approximately 22 degrees Celsius.

c. With CR205 cabled to your PC's COM port, connect with PakCom, click on Monitor Data Tables, enter an Update in Seconds of "2", and select the Public data table. The AirTemp_C that you are measuring or simulating will be displayed as below.

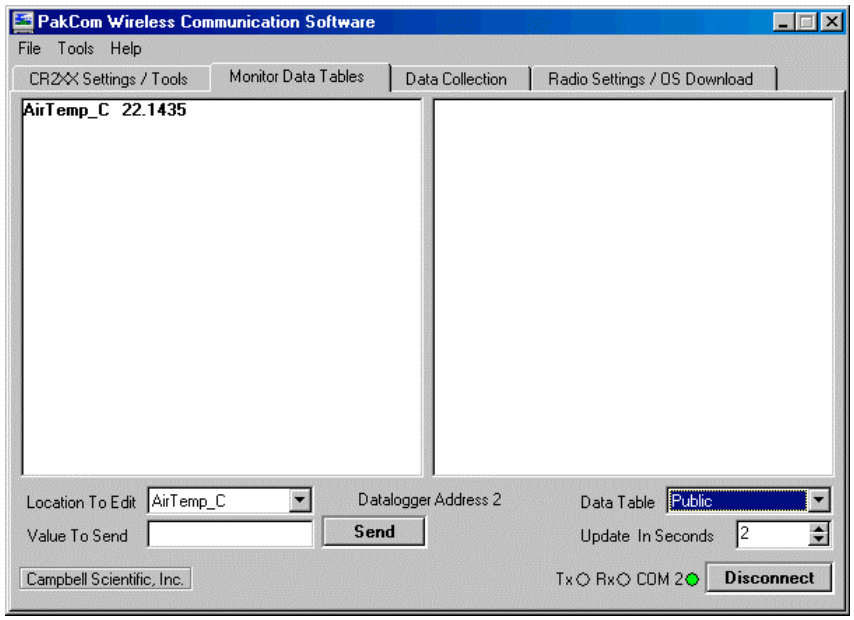


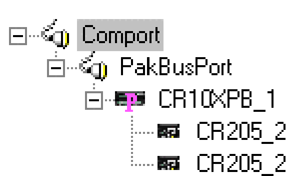
FIGURE 5. Monitoring CR205

Step 10 – Set up LoggerNet Device Map

a. Close PakCom and any other program that might be using your COM port.

b. Open the LoggerNet Toolbar and click the Setup button.

c. Create this device map:



d. Add a device to the map by highlighting the last device and then right-mouse click to see available options. Highlight and press the "Rename" button to edit a default name.

e. Highlight ComPort_1 and select a COM port that is available on your machine. Communications Enabled should be checked and Extra Response Time should be zero.

f. Highlight PakBusPort and make the Beacon Interval = 0 (apply it).

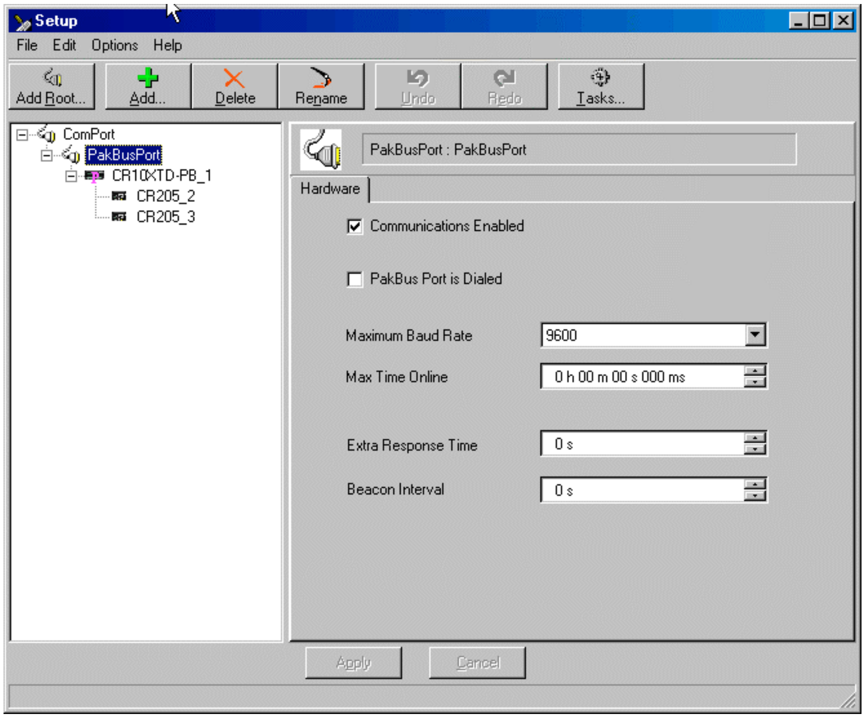


FIGURE 6. PakBusPort Settings

g. Highlight CR10XPB_1 and verify the settings shown.

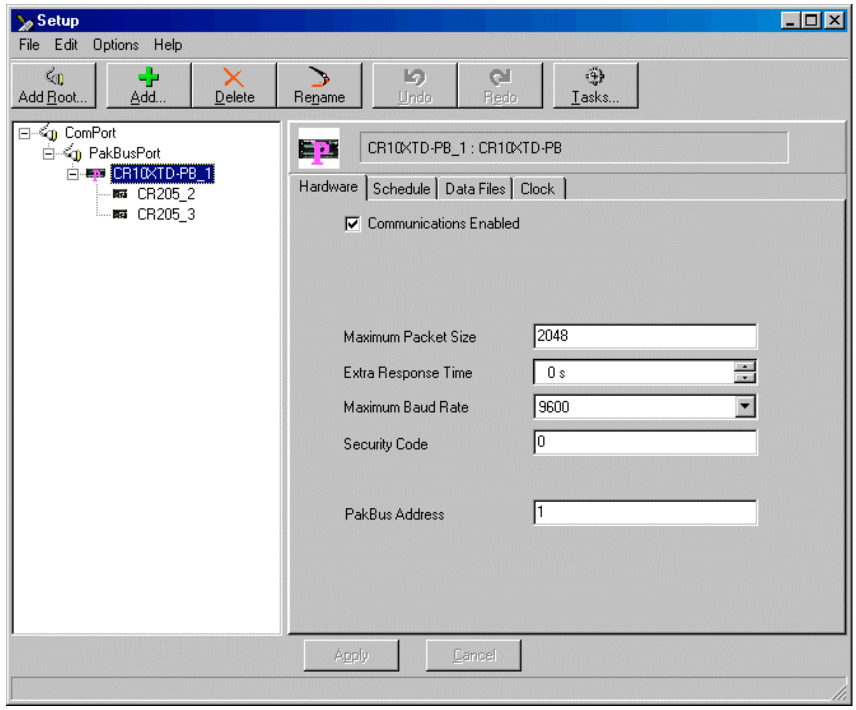


FIGURE 7. CR10XPB_1 Settings

h. Highlight CR205_2 and select PakBus Address 2.

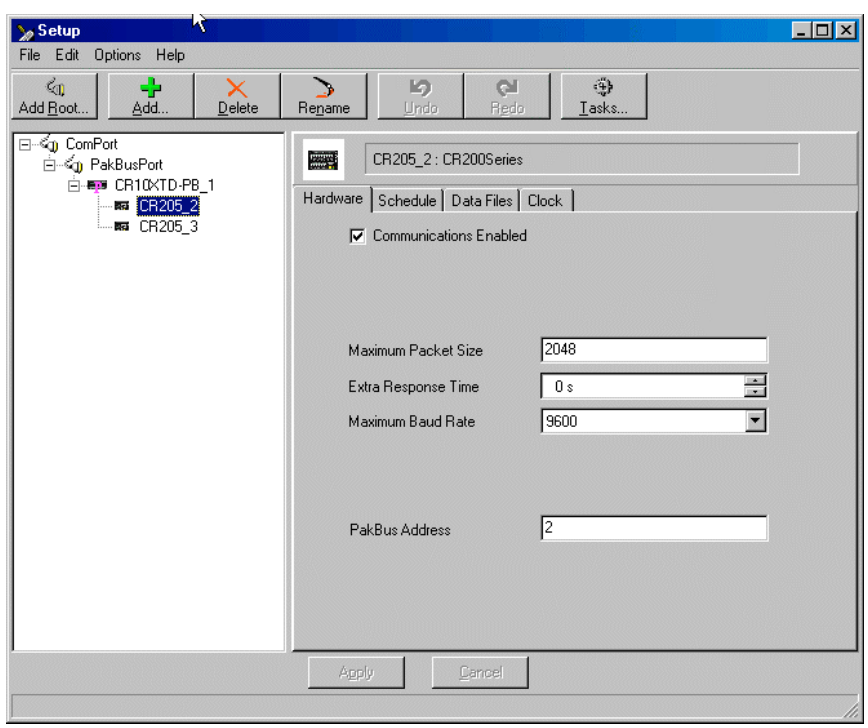


FIGURE 8. CR205_2 Setup

i. Highlight CR205_3 and change PakBus Address to 3.

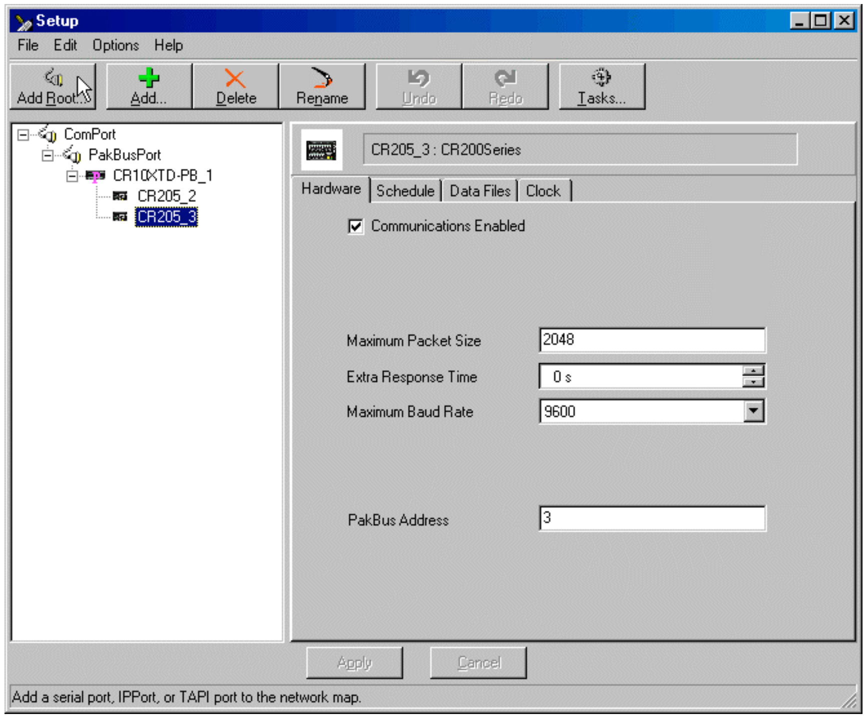


FIGURE 9. CR205_3 Setup

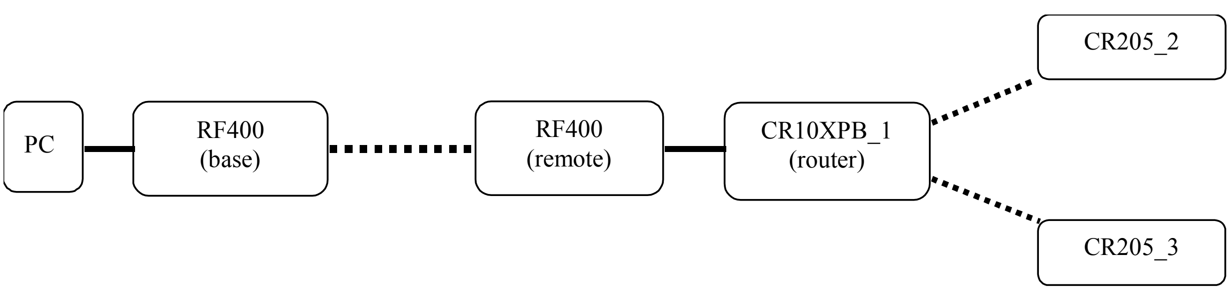
Step 11 – Set up Network Hardware

a. Verify the following:

- ✓ Antennas are on RF400s and CR205s.
- ✓ 12V supply is connected to base RF400
- ✓ Base RF400's RS-232 port is attached by SC12 cable to PC COM port

b. Cable the 'router' RF400's CSI/O port to the CR10XPB.

c. Place CR10XPB/RF400 and CR205s within range of each other.



Step 12 – LoggerNet Connect to Dataloggers

a. Highlight the CR10XPB_1 device in LoggerNet Connect Screen's Station window.

b. Click the Connect button.

c. Open Status Monitor on the LoggerNet Toolbar and highlight ComPort_1. Click View I/O and size/position the Low Level Log window to view I/O occurring between LoggerNet and the CR10XPB.

On the left of the window the "T"s represent data LoggerNet transmitted and the "R"s represent data LoggerNet has received. Notice that the PakBus packets begin and end with a "BD". A constant stream of Clock Check/Set Command and Clock Check/Set Response packets (03 and 83, see 9th hex digit pair beyond starting 'bd") will appear unless the "Pause Clock Update" box is checked.

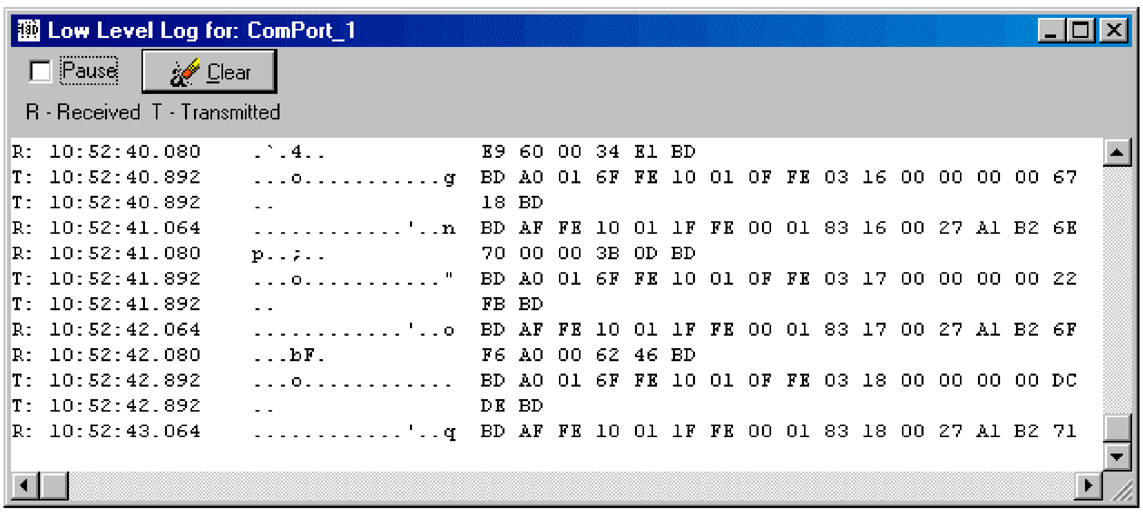


FIGURE 10. LoggerNet Low Level Log

d. In Connect Screen, Disconnect from CR10XPB_1 and highlight CR205_2.

e. Click the Connect button.

f. Update table definitions in Setup by highlighting CR205_2, clicking the Data Files tab, and clicking on the Get Table Definitions button. After a few seconds the message, "Received table definitions for CR205_2" will appear.

g. Set up a Numeric Display to view CR205_2 variable values. Click on Connect Screen\Data Displays\Numeric button "1". A Numeric Display window should now be open. Click the Add button. In the Add Selection window highlight "Public." The public variables for the CR205_2 program will be listed in the right-hand panel. Highlight them and Paste them to the Numeric Display. You should now see: RecNum, TimeStamp,

and AirTemp_C. The values associated with these will update every 10 seconds which is the scan rate of TEMP_PROBE.CR2.

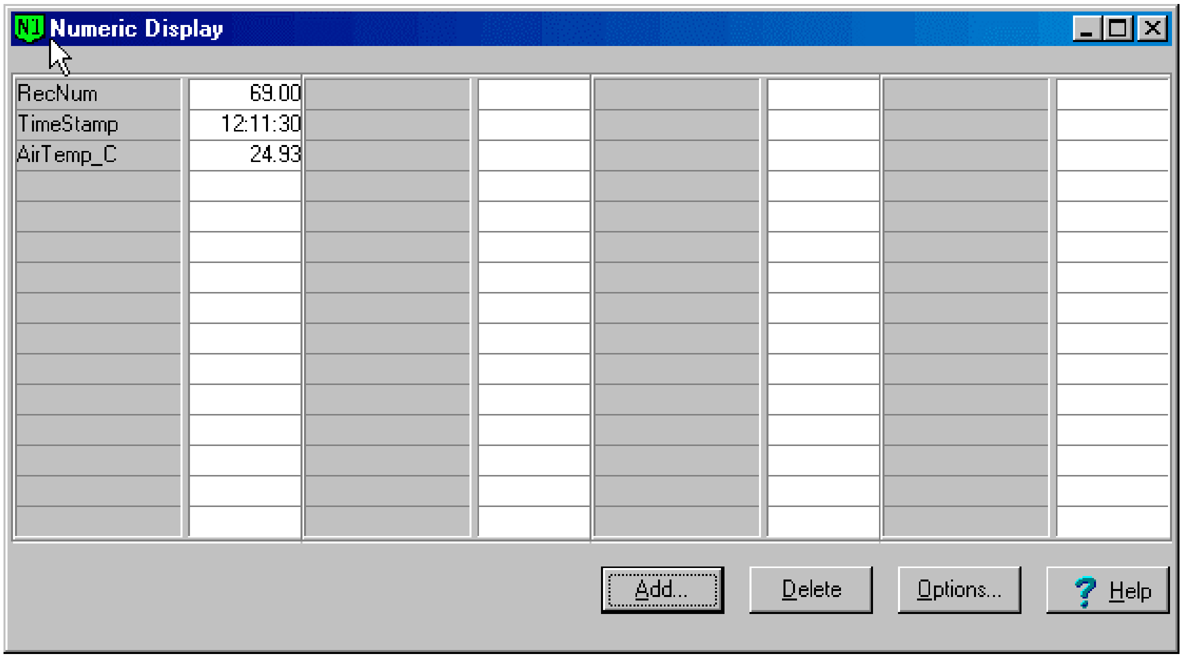


FIGURE 11. Display of CR205_2 Variables

h. Connect to and view AirTemp_C, etc. in CR205_3.

i. Your PakBus Network is now functioning. You can collect data, send a different program, or monitor any of the three PakBus dataloggers.

j. With LoggerNet not beaconing, you must connect to CR10XPB_1 to discover it. CR10XPB_1 has CR105_2 and CR205_3 in its Potential Neighbor list so it will quickly discover them and make them neighbors. LoggerNet needs to connect using a static route (from the device map) the first time it connects to CR10XPB_1. Once LoggerNet communicates with the CR10XPB it creates a dynamic route and keeps it in its routing table.

3. Choosing a PakBus Network Configuration

3.1 PakBus Overview

3.1.1 PakBus vs. Mixed-array

Mixed-array dataloggers include the older CR10, 21X, CR7 and CR500 and are programmed using Edlog, however, these do not support PakBus operating systems (OSs). The CR10X, CR510, and CR23X can be configured as mixed-array dataloggers but they also accept a PakBus OS because their OSs are stored in flash memory or Eprom and can easily be changed via CSI's OS download utility.

When should PakBus be used?

When adding to an existing system it may make sense to remain mixed-array, especially if your network must include non-PakBus dataloggers such as the 21X, CR7, etc., or if you rely on storage modules. However, communication equipment costs for PakBus systems are usually the same as for traditional communication and, although, like any other change, it takes some time and effort to learn the ins and outs of PakBus networking, the advantages are many.

Advantages of PakBus:

- Table based final data storage with more conventional strings for date/time stamps.

- Datalogger instructions that are the same as array-based except for the new P84 Data Table instruction and P190, P193 and a handful of associated instructions for efficient 'wireless sensor' cluster communications.

- Concurrent communications.

  - ✓ allows keyboard display to access datalogger while connected.

  - ✓ allows two PCs access to a particular datalogger at the same time.

- Distributed, intelligent route management.

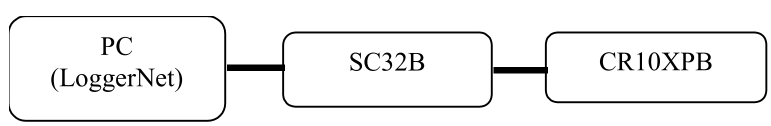
  - ✓ allows peer-to-peer communications (i.e., datalogger can easily exchange Input Locations or variables with any other network datalogger).

  - ✓ allows you to change communication routes remotely and let the network relearn the routing on its own.

  - ✓ allows you to program automatic back-up routes.

3.1.2 Minimum PakBus Network

The simplest PakBus network consists of a PC running LoggerNet with a COM port connecting via an SC32B to a PakBus datalogger.



- LoggerNet has the default PakBus Address of "4094".

- LoggerNet's Setup device map:

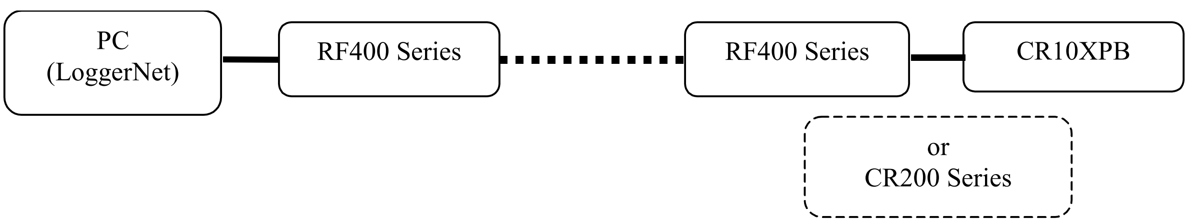
```
ComPort_1
  PakBusPort
    CR10XPB
```

NOTE PakBusPort is not an actual device, rather, it tells the LoggerNet server to use PakBus communication protocols with any device below it.

- Configure the CR10XPB in the device map the same as the datalogger's PakBus Address.
- When you Connect to CR10XPB the connection works because CR10XPB is in LoggerNet's device map which constitutes a "static route" to CR10XPB.
- You can send a program written in Edlog for the CR10X-TD (a program for a PakBus datalogger is programmed the same as for a Table Data datalogger).
- You should avoid sending a program with PakBus Settings that would change the PakBus address of the datalogger. You would find that by sending such a program it would appear that the program send had failed and you could no longer connect with that datalogger in LoggerNet. Should this happen, you could connect once again after changing LoggerNet Setup's device PakBus Address to the new address, but intentionally doing this as a means of changing the PakBus address is not recommended.

3.1.3 Minimum + RF400 Network

You can add RF400 Series communications to the 'minimum network' like this:

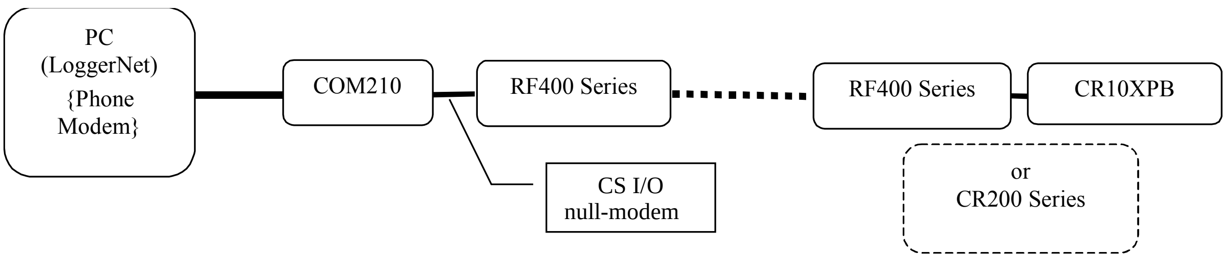


- The LoggerNet Setup device map stays the same.
- CR10XPB settings remain the same.
- RF400 Series radios in the network need to be set up:
    - ❑ Same Radio Address
    - ❑ Same Net Address
    - ❑ Same Hopping Sequence
    - ❑ Same Standby Mode (normally "< 24 mA always on"); other Standby Modes are possible, for example, the default "< 4 mA ½ sec cycle" or "< 2 mA 1 second cycle" work in some situations if power budget is an issue. Each RF400 and CR200 Series radio in the network should have the same Standby Mode (Radio Power Mode) setup.

- The PC's RF400 needs its own 12 VDC supply.
- This network behaves the same as the ‘minimum network’.
- You can install a Neighbor Filter in CR10XPB making it a ROUTER and add CR205s to the network. By so doing you would end up with the Quick Start network described in Section 2 of this guide.

3.1.4 Minimum + Phone Modem + RF400 Network

Adding a telephone modem link for the first hop:



- LoggerNet’s Setup device map now looks like this:

  ComPort_1
  PhoneBase
  PhoneRemote
  PakBusPort
  CR10XPB

- If the RF400’s Active Interface is configured for “COM2xx to RF400” a PS100 and an A100 will be needed to power the two peripherals and provide a CS I/O null modem. Or the RF400 Active Interface can be set for CSDC (7) and a PakBus datalogger connected along with the COM210.
- Alternatively, for TCP/IP communications substitute an NL100 for the COM210 and an IPPort in LoggerNet instead of the COM port.
- Multidrop communications could be used substituting a pair of MD485s for the PC modem and COM210. An RS-485 twisted pair cable connects MD485s. Additional MD485s with dataloggers can be added. Remember to give each datalogger a unique PakBus Address.

3.1.5 Interface/ Com Port Considerations

A PakBus datalogger can interface to a communications peripheral using the protocols listed below. The CR23XPB, CR200 Series, and NL100 have an RS-232 port.

- ***Modem Enabled***

  Non-CRBasic PakBus dataloggers support M.E. (modem enabled) communications (like non-PakBus dataloggers, only one M.E. device can be connected to a datalogger).

- ***CSDC and SDC***

  All PakBus dataloggers support CSDC (concurrent synchronous device communications). The CR10KD and COM310 can be used with PakBus dataloggers. Older SDC devices, such as the COM300, are not PakBus compatible.

- ***RS-232***

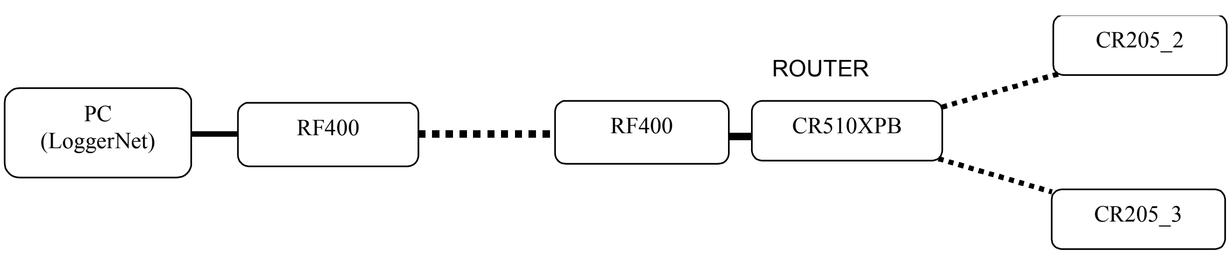
  The CR23X with PakBus OS supports PakBus communications on its RS-232 port. It can connect directly with the NL100. The CR200 series (some with built in radio) have an RS-232 port for direct connection to a COM port or NL100.

NOTE When you set up a communications peripheral for a particular port (Modem Enabled, CSDC 7, CSDC 8, or RS-232) the connected PakBus datalogger can auto-detect that port and connect with the peripheral for incoming *directed communications* to the datalogger. However, if the datalogger has a neighbor filter or beacon, there is a port configuration that is part of its neighbor filter or beacon settings which must match the communication peripheral's active interface (port) or the hello messages/beacons will not be sent.

For example, a CR10XPB router with neighbor filter must be configured for *D19 Port code of "17" if the attached RF400 is configured for "CSDC 7" Active Interface. Otherwise, the neighbor filter will not discover its assigned potential neighbors.

3.1.6 Router Considerations

With a PakBus rf (radio frequency) network you can add a router wherever you want, as many as you want, and the intelligent network will learn what you've added and incorporate it into routes wherever it provides a connection that didn't exist or a faster connection. It may also provide a backup connection for some links.



A router requires:

- A Neighbor Filter or Beacon setup in either the router or all of its neighbors to discover each other. If you have two in-range routers using neighbor filters, in order for them to discover one another you must list each of them as a potential neighbor in the other's neighbor filter.

- RF400 Series or other transparent communications devices such as spread spectrum radios produced by Freewave Technologies, Inc.

3.2 Approaches to Network Configuration

PakBus routing and buffering capabilities make possible multiple network topologies. For example, if an RF400 is used to communicate with a datalogger, which in turn communicates (through the same RF400) with more distant dataloggers, there are at least three basic ways to configure the network. Which model is best for your situation depends in part on your data collection and power budget requirements. The following trade-offs exist:

Configuration Trade-offs

one large file	VS.	a file for each station
faster data collection	VS.	slower data collection
larger remote power budgets	VS.	smaller remote power budgets

FIGURE 12. PakBus Configuration Trade-offs

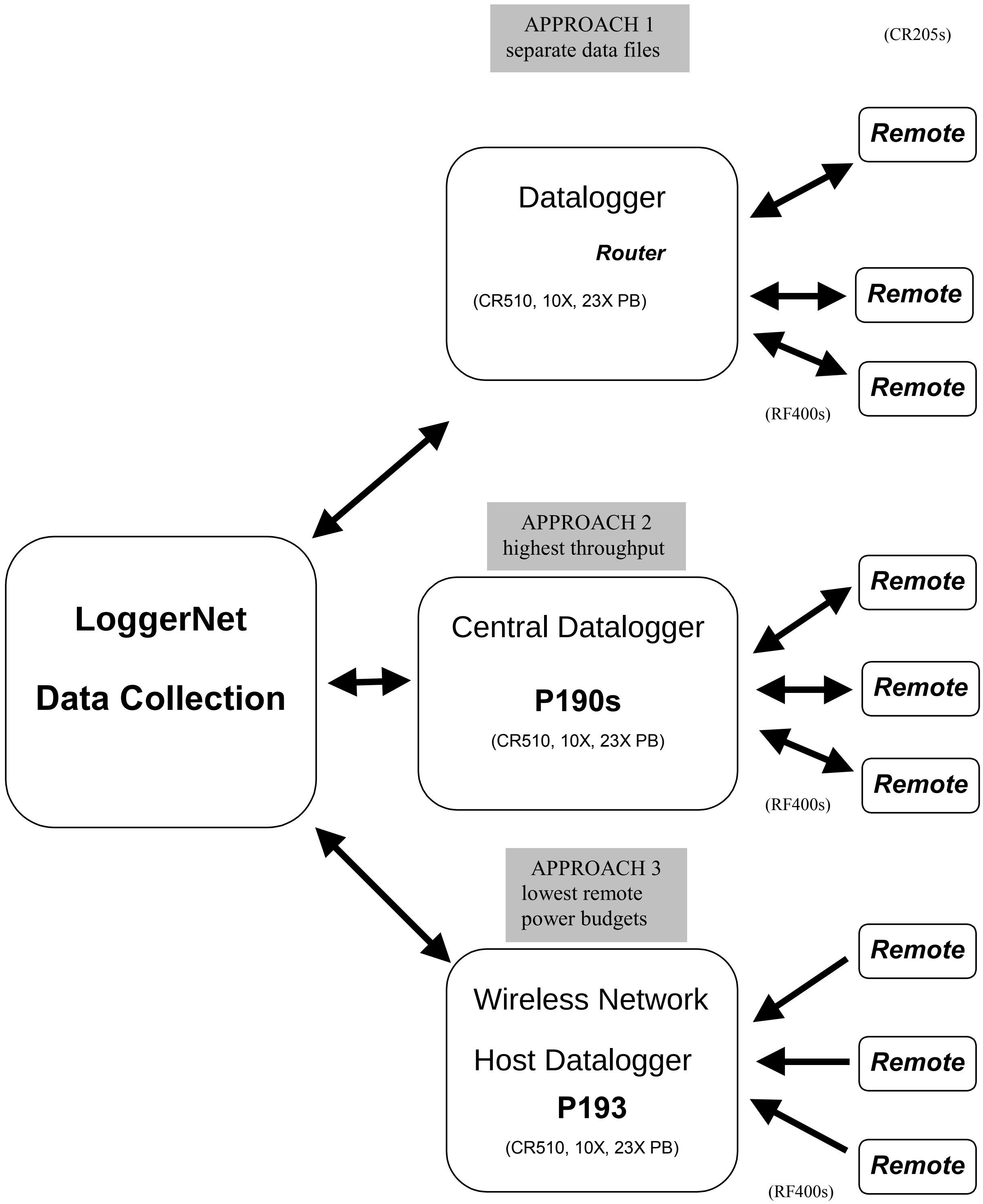


FIGURE 13. PakBus Data Collection Approaches

3.2.1 LoggerNet Direct (Approach 1)

LoggerNet communicates with each datalogger to collect historic data, resulting in separate files for each datalogger in the network. Each remote datalogger can have its own collection schedule.

3.2.2 P190 Datalogger (Approach 2)

A datalogger plays the role of concentrator using P190 instructions to obtain data values from the remotes. It processes the data and puts it into its own final storage for LoggerNet to collect to one file. This approach ignores historic data in the remote dataloggers.

For large networks this approach maximizes the speed of collection. Dataloggers can be collected as quickly as remote response allows.

3.2.3 P193 Datalogger (Approach 3)

The P193 assigns each remote an exclusive time slot to send data values. Thereafter, data transfers are initiated by the remote datalogger. This approach ignores historic data in the remote.

This reduces remote power budgets because radios only power up during their own time slot. However, P193 throughput and collection flexibility is less than with the P190 because of the fixed time slots.

3.3 Communications Choices

Many CSI communications devices support PakBus communications. A few do not.

3.3.1 Support PakBus Communications

- Wired

  SC32B
  MD485 multidrop RS-485 communications
  SRM5A short range asynchronous modems

- Phone

  COM210
  COM310

- Radio

  RF400 Series 100 mW spread spectrum radio communications
  MCC-545B 100 Watt meteor burst communications
  Freewave® 1 Watt spread spectrum communications (non-CSI device)

- IP (Internet Protocol)

  NL100 default router
  Redwing cellular digital packet data (CDPD)

- SAT HDR GOES

- PDA (Palm Handheld Device) using CSI's PConnect software

3.3.2 Don't Support PakBus Communications

- RF3xx radios/modems

- MD9

- COM300

- Storage modules

These can sometimes be used as a transparent link ahead of the PakBus network.

FIGURE 14. Some PakBus Communication Configurations

PC LoggerNet	COM Port	SC-32B	CR10XPB			
			CR10XPB RF400 Series (router)	RF400 Series CR10XPB (router)	CR200 Series	
		SRM5A	SRM5A	CR23XPB[1]		
				CR23X RF400 Series (router)	RF400 Series CR10XPB (router)	CR200 Series
		RF400 Series	CR200 Series			
			RF400 Series SRM5A	SRM5A SC932A CR10XPB		
			RF400 Series MD485	MD485 CR10XPB	MD485 CR10XPB	MD485 CR10XPB
			RF400 Series CR10XPB (router)	RF400 Series CR10XPB (router)	CR200 Series	
					RF400 Series CR10XPB (router)	CR200 Series
		Freewave®[2]	Freewave® SC105 CR10XPB (router)	Freewave® SC105 CR10XPB RF400 Series (router)	RF400 Series CR10XPB (router)	CR200 Series
			Freewave® MD485	MD485 CR10XPB	MD485 CR10XPB	MD485 CR10XPB
		Phone Modem	COM210 or 310 CR10XPB (router) RF400 Series	RF400 Series CR10XPB (router)	RF400 Series CR10XPB (router)	CR200 Series
			COM210 PS512M RF400 Series	RF400 Series CR10XPB (router)	RF400 Series CR10XPB (router)	CR200 Series
			Modem w/ RS-232	Null modem cable	CR200 Series	
		MD-485	MD485 CR10XPB	MD485 CR10XPB	MD485 CR10XPB	MD485 CR10XPB
	Ethernet	NL100	CR10XPB			
			RF400 Series	RF400 Series CR10XPB (router)	CR200 Series	
			MD485	MD485 CR10XPB	MD485 CR10XPB	MD485 CR10XPB
			Freewave®	Freewave® SC105 CR10XPB RF400 Series (router)	RF400 Series CR10XPB (router)	CR200 Series
		Cellular IP	Digital Modem (1xRTT or GPRS)	RF400 Series	RF400 Series CR10XPB (router)	CR200 Series
				SC932A	CR10XPB	
				CR23X		

* CR10XPB could be CR510PB or CR23XPB

[1] Uses RS-232 port. If you use CR10XPB, CR510PB, or CS I/O port of CR23XPB you must use SC105 interface

[2] Spread spectrum radios from Freewave Technologies, Inc.

3.4 Beaconing vs. Neighbor Filter Hello

Once you have chosen the collection approach and communication devices for your PakBus network, you need to provide a way for the PakBus devices to discover in-range devices. Some PakBus devices function as routers and, as such, are responsible to know the best route to every PakBus device in the network. The first step is for routers to discover their own neighbors. Next, they share their neighbor lists with all other routers in the network. End of the line "Leaf node" devices don't normally beacon or hello.

Discovery is done by:

- Neighbor Filter hello-ing
- Beaconing
- A combination of beaconing and hello-ing
- P190, P191, P192, P196, P198 or P224 peer-to-peer communications
- LoggerNet device map

The P193 instruction does not initiate communications and so cannot be relied upon to discover neighbors.

A device with neighbor filter has a potential neighbor list where the user lists the device addresses that the router attempts to discover by sending directed hellos. Other in-range PakBus devices are thus filtered out so they can't become neighbors. If you have two in-range routers using neighbor filters, in order for them to discover one another you must list each of them as a potential neighbor in the other's neighbor filter.

A beaconing device broadcasts to *all* neighboring PakBus devices. Any device within range, that has no neighbor filter to prevent it, will respond and become a "neighbor" to the beaconing device. If the beaconing device is LoggerNet, all in-range PakBus devices will respond (with or without neighbor filter).

Sometimes a simple way to discover other PakBus devices is to set up a beacon, however, a neighbor filter works better in most situations. This is particularly true of rf networks where beaconing can result in marginal links, whereas a neighbor filter approach for your routers forces them to use strong rf links (see Quick Start and Forced Routing sections).

LoggerNet is able to discover an in-range PakBus device at connect time using the 'static route' from its device map (no beaconing or hello filter required).

In addition to the discovery of neighbors, the above methods are also used to maintain ongoing lists of viable routes.

3.4.1 CR200 Series

CR200 Series devices, which don't already have a neighbor, respond to a beacon after waiting a random 0 to 1 seconds, by sending a hello-request

message back to the beaconing device. The beaconing device answers with a hello message to the CR200 Series device which returns a hello response message and they become neighbors. Discovery begins quickly for CR200 Series devices but it may take several beacons for all to be discovered depending on the quantity.

3.4.2 CR10X/CR510/CR23X PakBus Datalogger

Since beacons are packets of data, the more devices that beacon, the more significant the overhead in the communications channel. Also, in networks of more than a half dozen or so PakBus devices with RF400 Series radios, the random collisions/retries resulting from beacons and beacon responses can produce significant delays. If the number of PakBus devices and the size of data collections is small there will be a relatively low incidence of collisions/retries. With larger numbers of PakBus devices or higher data collection rates or amounts, the use of neighbor filters in routers to eliminate beacons can improve data throughput.

3.4.2.1 Beaconing

A CR10X/CR510/CR23X PakBus device waits a random 1 to 4 seconds after hearing a beacon before sending a hello response, so the probability is high that the response will get through.

Once they become neighbors, a beacon could collide with other communications taking place, so success depends upon the rate of beaconing and the rate of other communications. For example, 5 minute scheduled data collection and 1 minute beaconing should work if data collection takes less than 1 minute. If a beacon should happen to collide, it would probably be heard the next time.

3.4.2.2 Neighbor Filter Hello-ing

A router's neighbor filter discovers neighbors while reducing network collisions by eliminating beacons and only sending a hello message to listed potential neighbors. After discovery, neighbor filter will send more hello messages to an established neighbor only when normal communications does not occur within the communications verification interval × 2.5.

It should be noted that if one of the listed potential neighbor devices doesn't exist or fails, the hello-ing will occur at random intervals until either the failed device is removed as a potential neighbor or communications is restored. Removal can be done remotely by sending a program with new Edlog PakBus Settings. Removal could be done on-site using *D19 on a keyboard display.

3.4.3 Beaconing and Neighbor Filter FAQs

Q1: Will a CR10XPB which has no neighbor filter put a beaconing LoggerNet (LN) server in its neighbor list?
A1: Yes, whether or not the CR10XPB is in LoggerNet's device map.

Q2: Will CR10XPB with a neighbor filter put a beaconing LN server in its neighbor list?

A2: Yes, whether or not CR10XPB is in LoggerNet's device map provided LoggerNet's PakBus Address is ≥ 4000.

Q3: Will CR10XPB with no neighbor filter put a non-beaconing LN server in its neighbor list?
A3: Yes and No. Not initially. There must be communications first. Yes, after you connect once to CR10XPB it will put LN in its neighbor list and keep it there until 2.5 × the CVI times out.

Q4: Will CR10XPB with a neighbor filter put a non-beaconing LN server in its neighbor list?
A4: Yes and No. Not initially. There must be communications first. Yes, after you connect once to CR10XPB it will put LN in its neighbor list and keep it there until LN is closed or LoggerNet's router is reset.

Q5: Will CR10XPB, who has no neighbor filter and who hears a non-neighbor beaconing, send it a hello message and accept it as a neighbor?
A5: Yes, this method of discovery is often used.

Q6: Will a CR10XPB with neighbor filter who hears a non-potential neighbor beaconing send it a hello message and establish it as a neighbor?
A6: No, with the exception of a beaconing device with PakBus Address ≥ 4000 in which case Yes (see Q2).

Q7: Will beaconing CR10XPB-router or LN send out a beacon during data collection from the CR10XPB?
A7: Yes, they will send out beacons between packets during data collection.

Q8: Can a leaf node CR10XPB beacon?
A8: Yes, but normally a router closer to LN or LN itself should do any beaconing.

Q9: Can a CR205 (always a leaf node) have a neighbor?
A9: Yes, a leaf node has a single-neighbor 'routing table.'

Q10: Can a leaf node CR10XPB have a neighbor?
A10: Yes, it can have a neighbor, however, lacking a *D15 configured routing table, it can't display that neighbor.

Q11: Will a beaconing CR10XPB receive hello messages from a group of CR205s?
A11: Yes… more or less. To be precise, a CR205 that has no neighbor will, upon hearing a beacon, respond with a hello-request message which will cause the beaconing device to send a hello, to which the CR205 will respond making them neighbors.

Q12: Can LN(4094) serve as a router in a PakBus network?
A12: Yes, without special setup one node can communicate with another node through the LN router.

FIGURE 15. Beacon and Neighbor Filter Discovery

LoggerNet Configuration	Node 1 Configuration	Node 2 Configuration	Will They Become Neighbors?
beaconing	no neighbor filter no beaconing	---	yes
beaconing	neighbor filter	---	yes
no beaconing	no neighbor filter no beaconing	---	yes, but you must connect once
no beaconing	neighbor filter	---	yes, but you must connect once
---	no neighbor filter	beaconing	yes
---	neighbor filter	beaconing	no (unless Node 2's address ≥ 4000 or Node 2 is listed as a potential neighbor in Node 1's neighbor filter)
---	neighbor filter (Node 2 listed as potential neighbor)	neighbor filter (Node 1 not listed as potential neighbor)	no

Note: a node is a PakBus Device such as the CR10XPB, usually a router.

4. PakBus Device Setups

4.1 Dataloggers

4.1.1 Edlog Datalogger Setup

You can put the PakBus settings you need in your *.dld program using Edlog's Options \ PakBus Settings. Sending the program to the datalogger will configure those PakBus settings, overriding any settings already in the datalogger.

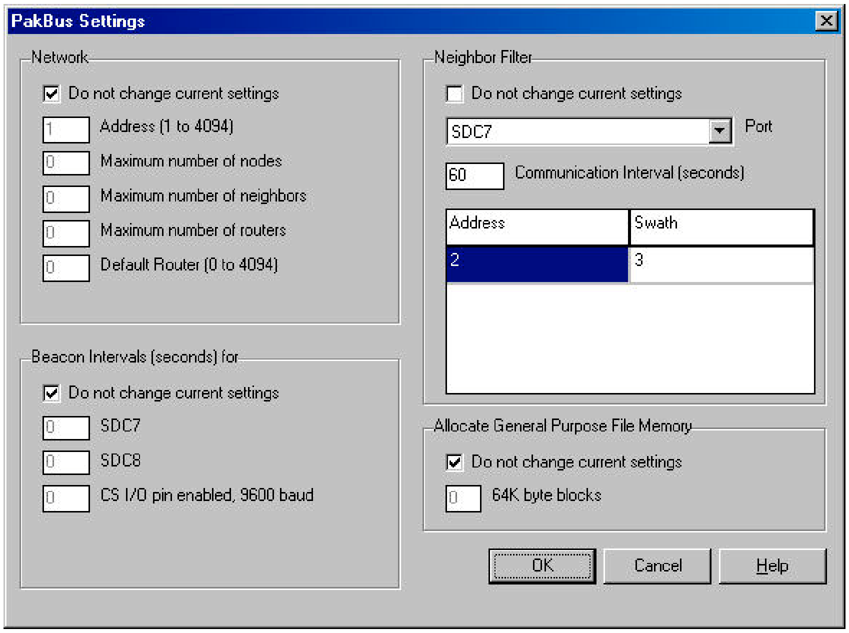


FIGURE 16. Edlog PakBus Datalogger Setup

If you don’t want to change a category, check the “Do not change current settings” box and when the program compiles in the datalogger, that area will remain the way it was. Be sure to save/compile the program after making the PakBus Settings changes before sending to the datalogger.

Notes

Zeroing a PakBus Settings “Communication Interval” will delete the selected port when the program is compiled and sent to the datalogger.

To insert a row of PakBus Addresses (Address / Swath), place cursor over that field, right-mouse click, and select “Insert Row.”

Configuring an existing port type will overwrite any previous settings for that port.

4.1.2 *D PakBus Setup for the CR10X, CR510 and CR23X

You can use Edlog \ Options \ PakBus Settings to configure a dataloggers’s PakBus settings or you can use a keyboard display to edit *D15, *D18 and *D19 to accomplish the same thing. *D17 is for viewing a datalogger’s routing table.

You can configure a datalogger’s PakBus settings anytime after downloading a PakBus OS, before or after sending the program. The *D settings become part of the program as soon as you press A. Pressing *0 saves the *D settings to flash. Current PakBus settings will persist through a new program send if the new program has no Edlog entered PakBus settings.

*4.1.2.1 *D15 Set PakBus Address and Create Routing Table*

Set PakBus Address

15:xxxx (1 to 4093, default = 1, be careful with numbers > 3999 (see Neighbor Filter section)).

NOTE If you make 01, 02, 03 values too small, you may not be able to connect to all network nodes.

Configure Routing Table

01 :xxxx Maximum nodes in PakBus network (enter 0 if a leaf node, non-zero if a router).

02 :xxxx Maximum neighbors to this PakBus Address (Maximum nodes must be non-zero).

03 :xxxx Maximum routers in PakBus network (Maximum nodes must be non-zero).

04 :xxxx Default Router Address (Maximum nodes must be non-zero; minimizes routing table size by specifying default router to another network branch).

CAUTION Collect any wanted data before typing *0 to save *D15 edits to flash, because final storage data is cleared!

*D15 settings allocate memory similar to *A. It is a good idea, when configuring them to leave 'room to grow.' Changing *D15 settings later on could result in loss of data.

If a device is unable to find a route to a destination in its routing table, and it has a *default router address* configured, it will send the packet out via the default router.

If in doubt as to how large to make max nodes, neighbors or routers in your network, you could guess a little high. As you can see from the table below, datalogger memory usage rises per the square of the number of routers.

*D15 Routing Table Datalogger Memory Usage**:**

Per node in network	16 bytes
Per neighbor	8 bytes
Per router in network	2 × routers × (routers – 1) bytes

For example - the Quick Start network in Section 1:

Number of nodes = 4 (LoggerNet is one node)   CR10XPB memory usage = 64 bytes

Number of neighbors to the CR10XPB router = 3CR10XPB memory usage = 24 bytes

Number of routers = 2 (counting LoggerNet)   CR10XPB memory usage = 4 bytes

4.1.2.2 *D17 View Routing Table

01 :xxxx	PakBus Address of destination node
01 :yyyy	via neighbor (router) with PakBus Address yyyy (yyyy is present only if a router is needed to access xxxx)
01 :zzzz	worst case response time in seconds (hop metric)
02 :xxxx	PakBus Address of destination node
02 :yyyy	via neighbor (router) with PakBus Address yyyy
02 :zzzz	worst case response time in seconds
...	
0n :xxxx	PakBus Address of destination node
0n :yyyy	via neighbor (router) with PakBus Address yyyy
0n :zzzz	worst case response time in seconds

A *D17 listing is added after 1 beacon response, held at least 1 beacon interval, and dropped ≈ 20 secs after no beacon response.

A *D17 listing is added after one hello response, held for at least 2.5 × the comms verification interval, and dropped if no response.

A “neighbor” is indicated in a *D17 Routing Table by a single PakBus Address in that window number. For example, if device 0003 is a neighbor to a datalogger, the datalogger’s routing table will look like this:

01: 0003   PakBus Address of destination node
01: 1.0000 worst case response time in seconds

The *D17 listing will change once the datalogger discovers a change in the neighbor list of any router in the network (including itself) that results in a change in the best routes to the nodes in the network. *D17 shows the current "best" routes; it does not show all routes even though the datalogger has access to all possible routes to any given node. If a router's neighbor list changes, all routers are notified and will recalculate their "best route" routing table based on the new information.

Although *D17 shows neighbors, in the case of multiple peripherals it does not show the ports you use to access the respective neighbors.

The P199 instruction copies the *D17 Routing Table to a block of Input Locations for transfer to another PakBus device.

4.1.2.3 *D18 Set Beacon Interval and Port Protocol

01 :xxxx Enter interval (seconds) for CSDC 7
02 :xxxx Enter interval (seconds) for CSDC 8
03 :xxxx Enter interval (seconds) for Pin Enabled, 9600 baud (required for "9600" NL100 PakBus connection)
04 :xxxx Enter interval (seconds) for RS-232 port, 9600 baud (CR23X only)

A value of "0" produces no beacons.

CAUTION Collect any wanted data before typing *0 to save *D18 edits to flash, because final storage data is cleared!

When you set up a beacon, its broadcast baud rate is equal to the *D12 configured port baud rate (0 is not allowed; if 0 is entered the baud rate defaults to 9600). If the user has not entered anything into *D12 then the baud rate defaults to 9600, even if auto baud rate detection has put the baud rate to something else.

4.1.2.4 *D19 Set Neighbor Filter

NOTE Configuring the *D19 Neighbor Filter requires that the *D15 Routing Table also be configured with max nodes, max neighbors, and max routers so the datalogger has a routing table in which to put neighbors discovered from *D19 potential neighbors.

19 :xx **Port**

17 CSDC 7
18 CSDC 8
0x M.E. (x is *D12 baud rate, typically "2" for 9600 baud; see datalogger manual for baud rates)
0x-- RS-232 for CR23X (x is *D12 baud rate, typically "2**"** for 9600 baud)

19 :xxxx **Comms Verification Interval** (Seconds)
Device sends a hello message if no comms by 2.5 × this interval

01 : xxxx Starting PakBus Address of Potential Neighbors (PNs)
01 : ss Swath of PN addresses

02 : yyyy Starting PakBus Address of PNs
02 : ss Swath of PN addresses
...
0n : zzzz Starting PakBus Address of PNs
0n : ss Swath of PN addresses starting with zzzz

Notes

If you have already set up the *D15 Router Table, as soon as you compile neighbor filter settings the datalogger starts sending hello messages. The potential neighbors soon appear in the *D17 Routing Table, enabling the datalogger to communicate with those devices (now neighbors) and, possibly, use them as routers.

If you have two in-range routers using neighbor filters, in order for them to discover one another you must list each of them as a potential neighbor in the other's neighbor filter.

You can change the neighbor filter using a CR10KD and *D19. Changes take place immediately as you edit and press A. Pressing *0 saves any edits to flash memory.

CAUTION Collect any wanted data before typing *0 to save *D19 edits to flash, because final storage data is cleared!

If you want to delete a setting, for example, a "17" port, type "0" over the 17 and press A to enter. Then type the desired port, if any, over the "0" and finish the configuration. Entering "0" or changing *D19's com port code resets the datalogger's neighbor list and routing table. This is a way to reset device discovery.

NOTE If you type, for example, "17" over "02" without first typing "0" and A to enter, you will end up with two ports (02 and 17).

Multiple Ports

You can *D19 configure multiple ports. For example, you can configure a 17 (CSDC 7) port with some PakBus Addresses, then configure a 02 (Modem Enabled) com port with other addresses. For example, a datalogger can be configured to support, as potential neighbors, links through a CSDC 7 RF400 and links through a CSDC 8 MD485.

To program two *D19 potential neighbor ports:

- Type *D19 A
- Type 17 A (for CSDC 7 port)
- Type 60 A (for comms verification interval)
- Type xxx A (1st PakBus Address)
- Type nn A (swath of nn addresses)
- Press B repeatedly until back to window displaying com port "17"
- Type 18 A to create a new port (CSDC 8)
- Type 60 A (for comms verification interval)

- Type yyy A (1st address)
- Type nn A (swath of nn addresses)
- Type *0 to save *D19 settings

The datalogger will now attempt to do hello exchanges with the PakBus Addresses you put in the potential neighbor list.

Before compiling, to add more potential neighbors to CSDC 8 com port:

- Type B repeatedly to get back to com port window
- Type 18 A (type over the top of com port code)
- Type 60 A (for comms verification interval)
- Type A A to get beyond existing PakBus Addresses and swath
- Type xxx A (1st address)
- Type nn (to add a swath of nn potential neighbors)
- Type *0 or *6 to compile settings

After compiling, to add more potential neighbors to CSDC 7 com port:

- Type *D19 A
- Type 17 A
- Type 60 A (for comms verification interval)
- Type A A to get beyond existing PakBus Addresses and swath
- Type xxx A (1st address)
- Type n A   (to add a swath of n)
- Type *0 or *6 to compile settings
  (datalogger does not combine consecutive additions into a single swath)

4.1.3 CR200 Series PakBus Setup

Use PakCom software to accomplish CR200 Series setup. Setup involves configuring PakBus Address, radio Hop Sequence, Network and Radio Addresses, and power settings. See Quick Start section for detailed information on PakCom CR200 series setup.

4.2 LoggerNet PakBus Setup

4.2.1 PakBusPort

A PakBus port must be set up in the device map prior to adding a PakBus datalogger. In order to communicate to a datalogger via another datalogger from LoggerNet's Connect window (i.e., communicating with a CR205 via a CR10XPB) the datalogger through which you will be communicating must be set up as a router. This is done by entering the number of remote dataloggers to which the router will be connecting in *D15, parameter 2.

4.2.2 PakBus Port is Dialed

Typically, a PakBus port is always active (open) so that any incoming packets that are transmitted can be received. When this check box is enabled, the PakBus port is only active when communication is taking place with the device.

If a phone modem or RF400 Series radio is represented in the Setup map, the port is 'dialed' regardless of this setting. Otherwise, other devices or software would not be able to use the PC's phone modem, and charges might be incurred for keeping the phone link active.

You will want to enable this option if there is any reason that you will need to use the root communications port for other devices or software, regardless of whether a phone modem is being used (e.g., if the COM port is also being used to communicate with a Storage Module using SMS).

5. Communication Device Setups

5.1 RF400 Series

Most PakBus networks involve RF400s even though they are often not represented in the LoggerNet device map. The rule-of-thumb for using RF400s in a PakBus network is that all RF400s in the network be configured exactly the same, with one exception. The RF400's Active Interface should be suited to the device's capability and according to any other devices connected to the PakBus device.

RF400 Series can be configured using the PakCom software that ships with them or LoggerNet's Terminal Emulator.

5.1.1 Active Interface

The preferred port type for PakBus communications is CSDC because it uses less datalogger CPU resources than M.E. If the datalogger has a modem enabled device attached, the RF400 must be configured for CSDC. If there is already a CSDC 7 device connected, change the CSDC setting to 8 in PakCom's Radio Settings \ OS Download, I/O Terminal. This is done in the RF400's Advanced Setup, Interface Parameters.

5.1.2 Addresses

All PakBus network RF400 Series should be configured with the same Network Address and Radio Address. PakBus devices that are in-range 'hear' every packet transferred, but they only respond to the packets addressed to them. Default addresses usually work. If you are in a busy RF400 area, you can reduce retries by choosing non-default Network and Radio Addresses.

5.1.3 Hopping Sequence

All PakBus network RF400 Series should be configured with the same Hopping Sequence (sometimes shortened to hop table).

If activity on the RF400s' green LED indicates substantial same-hop-table activity from a nearby network, you should consider changing to a more free hopping sequence to reduce retries.

5.1.4 Standby Mode

Normally the RF400 is configured for "<24 mA always on." Other Standby Modes are possible, for example, the default "< 4 mA ½ sec cycle" and "< 2 mA 1 sec cycle"standby modes work for some configurations if power budget is an issue. Response times are a little better when RF400s are configured for "<24 mA always on" since the RF400 receiver is always active instead of cycling in and out of sleep mode. Each RF400 and CR200 Series radio in the network should have the same Standby Mode (Radio Power Mode) setup.

5.2 MD485

The MD485 allows multiple PakBus devices to be cabled together on an RS-485 bus (one twisted pair with shield) instead of using radio communications. Maximum range is around 4000 feet of total cable length.

5.2.1 Communication Mode

In a PakBus network the Communication Mode must be configured as "PakBus Networking" to enable the low-level collision avoidance protocol.

5.2.2 Active Ports

The two ports to set active are RS-485 and either CS I/O or RS-232 according to the capability of the PakBus datalogger and your usage of the CS I/O port.

5.2.3 Baud Rates

This setting will usually be 9600 but can be greater if the datalogger allows.

5.3 NL100

Use Telnet or Terminal emulator to configure an NL100 (see user's manual). The NL100 can be used in transparent mode (TcpSer) where, for example, it's

RS-232 port is connected directly to an RF400 Series communicating with CR200 Series remotes (see Appendix B).

Alternatively, the NL100 can be configured as a PakBus device with a PakBus Address and beacons enabled:

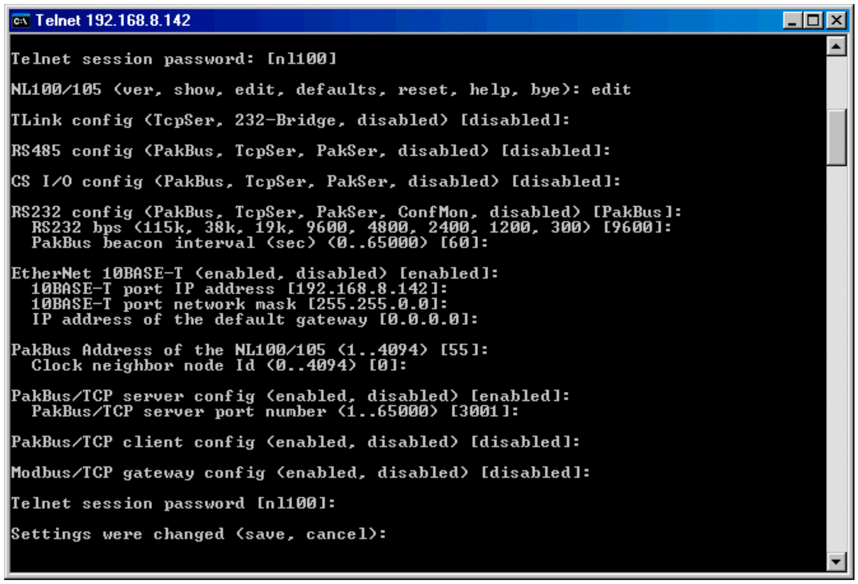


FIGURE 17. NL100 Configured as PakBus Device

In the above example, both the NL100's CSI/O port and RS-232 port are configured for "PakBus" mode.

5.3.1 CS I/O and RS-232

Can be "TcpSer" for transparent operation or "PakBus" (see figure above) to create a PakBus device.

5.3.2 Beaconing

If you create a PakBus node you can also create a beacon with programmable interval.

5.3.3 PakSer

Port configuration for a non-PakBus datalogger in the network.

6. Datalogger PakBus Instructions

6.1 CR10X, CR510 and CR23X Dataloggers

Following are summaries of Pxxx instructions pertaining to PakBus networks. For more information, refer to Edlog Help or datalogger manual.

6.1.1 P190 – Send or Get Input Locations

Purpose:

To send or get Input Location data from a non-CRBasic datalogger in a PakBus network. Best for large networks in terms of getting data from many remotes the most quickly since the P190 can execute as fast as responses allow. P190 can also act as a Modbus master and issue commands to Modbus devices.

NOTE In order to establish P190 communications with a CR205, the P190's 1st parameter must agree with the attached RF400's Active Interface. For example, if the RF400 is set for CSDC 7, then P190 Parameter 1 must be 17 (CSDC7 and SDC7 are equivalent).

Remember to *D15 configure a Routing Table for max nodes, max neighbors, and max routers so the datalogger can function as a router. A good starting point is: *D15, xxxx, 5, 5, 5, 0.

Example:

```
6:  PakBus - Get/Send Locations (P190)
   1:   17           SDC 7                         ; Associated RF400's Active Interface (same as CSDC 7)
   2:   2            Address                       ; CR205's PakBus Address
   3:   26           Get Value                     ; For CR205, 21,22 are for remote CR10XPB w/P196
   4:   0000         Security
   5:   0000         Remote Loc/Coil/Register
   6:   1            Swath
   7:   3            Local Loc [ CR205_Temp_Dat ]
   8:   5            Result Code Loc [ StatusOfGet1   ]
```

If this instruction is used to retrieve a value or set a value in the remote datalogger's public table an Instruction 63 or 68 is required following the P190 to specify the variable name to be accessed (see Quick Start). No terminator character is needed in the P63/P68. The variable to be retrieved/set must be declared a *public* variable.

Index Parameter 3 (--xx) to delay execution of subsequent program instructions until the datalogger receives a valid response or error from the remote. Otherwise, the PakBus command is queued, the datalogger proceeds to the next instruction, and the communications are handled later when the remote replies.

<u>Result Codes</u>

0	Successful
>0	Initial attempt failed (value indicates the number of retries)
-99999 (or -INF)	The datalogger does not know the route to the remote datalogger.

The P190 can discover a neighbor. If the datalogger is neither beaconing nor hello-ing, then the P190 will hello the node even if it does not have the route yet by creating a static route from the specified address. Once it communicates,

then the CR205 becomes a neighbor to the datalogger which it then advertises to network routers including LoggerNet.

A P190 addressed device doesn't have to be a neighbor to the datalogger. The route to the address specified in P190 can be through a neighbor router.

If a default router is configured in *D15, the datalogger will send the packet to that node rather than assuming that the P190 Parameter 2 address is a neighbor.

A failed P190 will retry 4 times at intervals equal to the response time indicated by the hop metric. When it fails, and if the remote is a neighbor, it will treat the neighbor as if the CVI has expired, which will in turn trigger 4 hello attempts to try to reestablish the device as a neighbor. If this fails, the neighbor is removed and subsequent P190 instructions will use the 'static' route specified in the P190 Parameter 2 Address.

6.1.2 P192 – PakBus Message

Purpose:

To send a clock report, reset a routing table, or say 'goodbye' to a datalogger in a PakBus network.

This instruction does not work with the CR205.

Example:

```
2:  PakBus - Send Message (P192)
   1:    00              Local DL Port
   2:    0000            Remote PakBus Address
   3:    00              Message Type
```

Parameter 3 Codes:

2	Clock Report - sends the current time to the remote device. A P195 in the remote datalogger will set the datalogger's clock using the time received.
12	Reset Routing Tables - resets addressed device's routing table
13	Goodbye - removes the sending datalogger from addressed device's neighbor list (and therefore all links to the sending datalogger).

6.1.3 P193 – Wireless Network Master (Host)

Purpose:

Prepares local datalogger to send and receive Input Location data from a number of wireless sensors (CR200 series or non-CRBasic dataloggers running P196) in a PakBus network. P193 assigns a time slot to each remote and the actual data transfers are initiated by the wireless sensors. This is the lower average current drain scheme since the CR200 series datalogger need only be active when it transmits in its time slot and can sleep the rest of the time.

Example:

```
9:  PakBus - Wireless Network Master (P193)
  1:   10             Number of Remotes
  2:   1              First Remote Address                ; 1st CR205 "Datalogger Address"
  3:   0              Time Into Transmit Interval (sec)
  4:   60             Transmit Interval (sec, 0 = use execution interval)
  5:   3              Transmit Delay Between Remotes (sec)
  6:   4              Swath to Receive                    ; Number of data values each CR205
  7:   11             First Loc for Data Received [ DataRcvd_1    ]
  8:   5              Swath to Send
  9:   21             First Loc to Send [ HostBattr_1   ]
  10:  3              Result Code Loc [ Response      ]   ; RCV'd, -1 = successful
```

According to the Number of Remotes parameter, you must assign Input Location groups to accommodate Data Received and Loc to Send. For example, with 10 remotes and Swath to Send = 5 you would use Input Loc Editor to assign 50 Input Locations with appropriate labels for data to be sent. With Swath to Receive = 4 you would assign 40 Input Locations for data to be received.

The remote will keep its clock synchronized by retrieving the host's clock and setting its own with each communication with the datalogger during their time slot.

Once set up, P193 does not initiate any communications but responds to any port (CSDC 7, CSDC 8, or M.E.) that receives a SendGetData message from a CR200 Series or non-CRBasic datalogger's P196 instruction.

Result Codes

When the datalogger receives a message from one of its remotes, the corresponding Result Code Location is set to -1. When Instruction 193 is executed, the Result Code Location is incremented by 1. Therefore, if communication is successful, the Result Code Location will be 0 after the execution of Instruction 193. If data transfer is unsuccessful, the Result Code Location for the remote that failed will be incremented, and will continue to increment with each failed attempt.

Multiple Instruction 193s can be used in a program to configure up to four different groups of dataloggers/wireless sensors.

6.1.4 P194 – Time Until Transmit

Purpose:

Used in conjunction with a conditional statement to determine how many seconds remain until the remote datalogger's Wireless Remote Instruction (P196) should be executed to initiate the next scheduled communications with a master datalogger.

Example:

```
4:  PakBus - Seconds Until Transmit (P194)
   1:    0003            Input Loc with Seconds Until P196 Transmit [ ____ ]
```

If the host has not been heard (using P193), then the Time Until Transmit is set at a random time slot within a 1 minute interval.

6.1.5 P195 – Set Clock from Address

Purpose:

Sets local datalogger's clock from a remote datalogger's clock after the remote datalogger executes a P192 to send a Clock Report.

Example:

```
5:  PakBus - Use Remote Clock Report (P195)
   1:    0011            Address
```

The single parameter is the address of the remote datalogger expected to send Clock Reports.

6.1.6 P196 – Wireless Remote

Purpose:

Used in conjunction with P193 to set up a remote datalogger to act as a wireless sensor/controller (like a CR205) in a PakBus network communicating in a time slot dictated by the host datalogger during first successful communication. Remote uses P194 in conditional statement to time execution of its P196 transferring data to host.

Example:

```
6:  PakBus - Wireless Network Remote (P196)
   1:    17              Local DL Port
   2:    0006            Master's PakBus Address
   3:    0000            Security
   4:    05              Swath to Receive from Master
   5:    0011            First Input Loc for Data Received [ Host Data_1 ]
   6:    04              Swath to Send to Master
   7:    0031            First Input Loc to Send [ Local_1 ]
   8:    0041            First Comms Status Result Code Loc [ Result_1 ]
```

The master's P193 automatically handles clock synchronization with the P196 remote.

6.1.7 P197 – Force Route through Address

Purpose:

To enter a PakBus Address into a datalogger's routing table forcing the first hop where user wants to avoid using the auto-routing broadcast.

Example:

```
7:  PakBus - Force Route (P197)
   1:    17              Local DL Port
   2:    0005            Neighbor (Address of first hop)
   3:    0010            Destination PakBus Address (same as neighbor if only one hop)
   4:    01              Hops to Destination Node
```

6.1.8 P198 – Set Setting

Purpose:

Program control instruction to configure the Port Setting in a PakBus datalogger. A P63 or P68 should follow with the value for the changed setting. If the line 2 Address is the PakBus Address of the datalogger itself, P198 can change its own setting.

Example:

```
8:  PakBus - Set Setting (P198)
   1:    17              SDC 7
   2:    0012            Address
   3:    3               Result Code Loc [ Result        ]

9:  Extended Parameters (P63)
   1:    02              Option
   2:    00              Option
   3:    00              Option
   4:    00              Option
   5:    00              Option
   6:    00              Option
   7:    00              Option
   8:    00              Option
```

Changes PakBus Address 12 port from 17 (CSDC 7) to 2 (ME, 9600 baud)

Result Codes:

1. –1001 Attempted setting is read only
2. –1002 Out of space in the remote
3. –1003 Syntax error
4. 0 Success
5. >1 Number of communication failures

6.1.9 P199 - Routing Table to Input Locations

Purpose:

Used for trouble-shooting; instruction stores the datalogger's routing table in Input Locations.

Example:

```
3:  PakBus - Routing Table Information (P199)
  1:    0011            First Loc [ _______________ ]
```

Three pieces of information are stored:

1 PakBus Address of the destination node
2 PakBus Address of the neighbor used to route to the destination node (this line not present if destination is a neighbor)
3 Maximum response time metric (in seconds)

Set aside Input Locations equal to the maximum number of routes you expect to see × 3 (node, neighbor, route metric) + 1 extra for the –1 terminator of the list. For example, if you expect a maximum of 6 routes, then 6*3 + 1 = 19 Input Locations.

6.1.10 P224 - Send Table Definitions

Purpose:

To transfer Table Definitions to LoggerNet under datalogger program control as datagrams per BMP5 Message Type 0x89. Can be used for one way media like Goes satellite. Other purposes pending.

Example:

```
P224:  Send Table Definitions
  01:  17              Local DL Port
  02:  0003            PakBus Address;  destination
  03:  4               Table Number:  0 selects all final storage data tables
                                      100 selects the current data table
```

6.2 CRBasic PakBus Dataloggers

The CR205 is a CRBasic programmed, PakBus compatible datalogger. Soon to come is CSI's full featured, CRBasic programmed, PakBus datalogger - the CR1000.

6.2.1 SendGetData

Purpose

The SendGetData instruction is used in remote CR200 Series dataloggers to send an array of values to the host datalogger, and/or retrieve an array of data from the host datalogger. A TimeUntilTransmit instruction is required to control the timing of the SendGetData instruction allowing it to transmit in the time slot assigned by the host P193 instruction. For a more complete program example, see Appendix F.

Example

```
Public TimeUntilTx
…..

TimeUntilTx = TimeUntilTransmit(1)   ' 1 = RF, 2 = RS-232
…..

If TimeUntilTransmit(1) = 0 then     ' time to communicate
        SendGetData (Response, Control(),Measurements(),1,1,1,00000)
Endif
```

Syntax

SendGetData (ResponseDest, GetData, SendData, Port, HostAddr, RouterAddr, Security)

Remarks

The host datalogger must be a CR10XPB, CR510PB, or CR23XPB datalogger (CR200 to CR200 data transfer is not supported).

Parameters

- ResponseDest

  The ResponseDest parameter is the variable in which a response code for the transmission will be stored. The response code indicates whether or not the transmission was successful. The codes that can be returned are:

Code	Description
0	Successful
-1	Response received but permission denied
-2	Response received but insufficient resources
1, 2..n	The number of timeouts waiting for a response

- GetData

  The GetData parameter is the variable or variable array in which the array of data from the Host datalogger will be stored. If a 0 is entered, no values will be stored.

- SendData

  The SendData parameter is the variable or variable array that will be transmitted to the host datalogger. If a 0 is entered, no data will be transmitted.

- Port

  The Port parameter is used to specify the CR200 Series com port that will be used to communicate with the remote device (host P193 device). Enter a numeric code:

Code	Description
1	RF
2	RS232

- HostAddr

  The HostAddr parameter identifies the PakBus address of the host datalogger.

- RouterAddr

  The RouterAddr is used to specify the address of a router device in the PakBus network that the datalogger must go through in order to communicate with the host datalogger. If no router device exists, enter the HostAddr.

- Security

  The Security parameter is the security code for the PakBus network.

6.2.2 TimeUntilTransmit

Purpose

The TimeUntilTransmit instruction is a function that returns the time remaining, in seconds, before communication with the host datalogger.

Example

(see SendGetData example program lines above)

Syntax

TimeUntilTransmit (Port)

Remarks

The TimeUntilTransmit value is derived from the time slot information that is sent by the host datalogger. If the host datalogger has not yet sent time slot information, this instruction will use a random time interval between 0 and 60 seconds until communications with the host is established.

Parameter

- Port

  The Port parameter is used to specify the CR200 Series com port that will be used for communications. Enter a numeric code:

Code	Description
1	RF
2	RS232

7. PakBus Concepts

7.1 PakBus Device

A PakBus device is a device with processor/software that allows it to communicate using CSI's PakBus packet protocol. Some examples of devices configurable as PakBus devices are: LoggerNet server, CR10X, CR23X, CR510, CR205, and NL100. The indicated dataloggers must have a PakBus OS to be a PakBus device, however, the CR205 and the soon-to-be-announced CR1000 are native PakBus devices. In this guide the term "device" is often used as an abbreviation for "PakBus device."

BMP5 (Block Mode Protocol 5) is a datalogger application protocol that rides on top of PakBus.

Each PakBus device in a network must have a unique PakBus Address in order for the network to function properly.

7.1.1 PakBus Address

The term "PakBus Address" is equivalent to PakBus ID, node ID and node address. The preferred term is PakBus Address. Every PakBus device needs a PakBus Address in order to receive, send, or route PakBus packets. LoggerNet's default PakBus Address ("PakBus Computer ID") is 4094 and its allowable range is 1 to 4094. PakBus devices have default PakBus Addresses (usually "1") that you can edit as needed.

7.1.2 Node

In PakBus networking a "node" is equivalent to a "PakBus device." The LoggerNet server, a PakBus datalogger, and an NL100 can all be configured as a PakBus node.

7.1.3 Leaf Node

A leaf node is a PakBus Device that cannot route packets. It can send and receive packets only. The CR205 is always a leaf node.

By default, the CR10XPB, CR510PB and CR23XPB are configured as a leaf node (not set up in *D15 as a router to forward packets toward other devices).

7.2 Packets

In a PakBus network, all communications take place via packet transfer. In LoggerNet's Status Monitor Low Level log you may see packets of two protocols: PakCtrl (network layer) and BMP5 (application layer). The PakCtrl packets handle low level network functions such as discovery and neighbor list passing among routers. The BMP5 packets are used for higher level application to application communications; for example, a BMP5 packet is used to send a program or to collect data from a datalogger. Both types of packet appear in the Status Monitor low level I/O log.

From the tables that follow, you can see that most packets occur in command/response pairs. For example, the BMP5 Collect Data Command (09) is sent to a datalogger. The datalogger responds by sending back Collect Data Response (89) packets with table data encapsulated until all the data is transferred to LoggerNet.

7.2.1 Packet Types

PakCtrl Packets (Protocol 0)		
Command	Response	Function
02	82	Clock Message
03	83	Echo
07	87	Read Device Settings
08	88	Write Device Settings
09	89	Hello
0a	8a	Send Neighbor List
0b	8b	Get Neighbor List
0c	---	Reset Router
0d	---	Goodbye
81	---	Delivery Failure Message

Note: the beacon has no defined packet type.

BMP5 Packets (Protocol 1)		
Command	Response	Function
03	83	XTD Clock Check/Set Command
04	84	Program File Download
09	89	Collect Data
0b	8b	Message Mode User I/O
0e	8e	XTD Get Table Definitions
14	---	1-way Data Message
15	95	Get Variable
17	97	Clock Command
18	98	Get Program Status
19	99	Table Control
1a	9a	Get Values
1b	9b	Set Values
1c	9c	File Download
1d	9d	File Upload
1e	9e	File Control
20		1-way Table Definition Message
22	a2	Wireless
a1	---	Please Wait

FIGURE 18. Decoded Packets

7.2.2 Example Packet Exchanges

PakCtrl Example Packet – Hello Command and Response

```
09:24:39.120 T bd a0 01 7f fe 00 01 0f fe 09 2d 01 02 00 3c 19   ..........-...<.
09:24:39.120 T bc dd bd                                          ...
09:24:39.136 R bd af fe 20 01 0f fe 00 01 89 2d 00 02 ff ff b8   ... ......-.....
09:24:39.152 R 15 bd                                             ..
```

BMP5 Example Packet – XTD Clock Check/Set Command and Response

```
09:24:40.105 T bd a0 01 6f fe 10 01 0f fe 03 2e 00 00 00 00 d2  ...o............
09:24:40.105 T 5e bd                                            ^.
09:24:40.137 R bd af fe 10 01 1f fe 00 01 83 2e 00 27 5b 2a 99  ...........'[*.
09:24:40.152 R 58 c0 00 d1 77 bd                                X...w.

Note: 9th digit and packet type are underlined
```

7.2.3 Packet Protocol Levels

Notice that some packet numbers are shared between protocols. To determine which it is, locate the 9th hex digit after the last leading "bd". If that digit is zero, the packet is a PakCtrl packet (Protocol 0). If a "1", it is a BMP5 packet (Protocol 1).

7.2.4 Headers

The two packet protocol levels of interest are PakCtrl (network level) and BMP5 (application level). When connected to a datalogger, the Status Monitor's Low Level I/O Log (COM port, View I/O) will show lots of Clock Check/Set Command and Response packets (BMP5 packets). Other packets appear as you perform functions in LoggerNet. Following are some example packets from the two levels.

PakCtrl Example Packet – Hello Command and Response

```
09:24:39.120 T bd a0 01 7f fe 00 01 0f fe 09 2d 01 02 00 3c 19  ..........-...<.
09:24:39.120 T bc dd bd                                         ...
09:24:39.136 R bd af fe 20 01 0f fe 00 01 89 2d 00 02 ff ff b8  ... ......-.....
09:24:39.152 R 15 bd                                            ..
```

BMP5 Example Packet – XTD Clock Check/Set Command and Response

```
09:24:40.105 T bd a0 01 6f fe 10 01 0f fe 03 2e 00 00 00 00 d2  ...o............
09:24:40.105 T 5e bd                                            ^.
09:24:40.137 R bd af fe 10 01 1f fe 00 01 83 2e 00 27 5b 2a 99  ...........'[*.
09:24:40.152 R 58 c0 00 d1 77 bd                                X...w.
```

7.2.4.1 Link State

The 1st hex digit after the last leading "BD" of a packet indicates the link state. Possible link states are:

1st Hex Digit	Link State
8	offline
9	ring
a	ready
b	finished
c	pause

The transmitted (T) *BMP5 Example Packet* above has the link state of "a" meaning that PakBus device "ffe" (LoggerNet server) is ready to talk to PakBus device 001 (a datalogger). The packet is addressed to device 001 from device ffe.

7.2.4.2 Two Level Packet Addressing

Examination of a packet header reveals four PakBus Addresses. Referring to the above *BMP5 Example Packet* (portrayed below) hex digits 2 – 4 (“001”) represent the neighbor PakBus Address to which the packet is going. Digits 6 – 8 represent the neighbor PakBus Address from which the packet came (“ffe”).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
bd	a	0	0	1	6	f	f	e	1	0	0	1	0	f	f	e

17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
0	3	2	e	0	0	0	0	0	0	0	0	d	2	5	e	bd

Digits 10 – 12 represent the ultimate destination PakBus Address. Digits 14 – 16 represent the source PakBus Address. In this example, since there are no routers, both address levels use the same PakBus Addresses. If there were routers in the path, you would see the packet assume different to/from neighbor PakBus Addresses as it traveled along but the source and destination would remain constant.

7.3 Neighbor

The term “Neighbor” has special meaning in PakBus networking. A neighbor to a PakBus device is another PakBus device with which it has recently communicated directly (without a router). In a *D17 routing table, neighbors are indicated by a single address xxxx (see *D Setup for the CR10X, CR510 and CR23X). If a second address yyyy is present then a route exists to the destination but at least one router is needed to get there. Before a PakBus device can consider another device to be a neighbor, they must successfully accomplish a hello exchange and periodically communicate in some fashion. Discovery is always two-way.

PakBus devices can discover a neighbor by:

- Sending a hello message to a *D19 Neighbor Filter potential neighbor and getting back a hello response
- Beaconing and doing a successful hello exchange
- Communicating via P190, P191, P192, P196, P198, P224 and doing a hello exchange
- Static route in LoggerNet or P190 datalogger communicating with a device followed by a hello exchange

Only a successful hello exchange can establish devices as neighbors.

For successful routing, network devices must maintain an accurate list of currently viable links. If a device has not heard from a certain neighbor within a period equal to the lesser of the two devices’ communications verification intervals × 2.5, it will initiate a hello-exchange. If the attempt fails, the device will try again. If unsuccessful after 4 such attempts, the neighbor will be

removed from its neighbor list. The same removal procedure will happen in the other device.

When a network route's neighbor list changes (either neighbor added or neighbor removed), it then shares that information with other routers. This behavior fundamentally drives the network routing process.

A device with a neighbor always has an associated communications verification interval. If a device has several neighbors, each link has a separate communications verification interval.

When one device establishes another device as a neighbor, it also learns the port through which it communicated with that neighbor. Although not displayed, the com port is included in the device's neighbor list.

Neighbor List

A PakBus device actively updates its list of neighbors. Regular normal communications between devices is sufficient to preserve their *neighbor* status. If a device hears a directed packet (addressed to the device) from a neighbor within 2.5 × the CVI, it will not need to otherwise communicate with that neighbor. If the time expires without hearing from the neighbor, it will initiate 4 hello message exchanges before giving up and removing the device from its neighbor list.

Whenever a router experiences a change in its neighbor list, it sends its new neighbor list to other *routers* in the network enabling them to update their routing tables.

Part of a neighbor list entry is the com port (02, 17, 18 or RS-232) through which to communicate with a neighbor. In order to send a packet to a neighbor you must have not only the correct PakBus Address, but also the correct port (matching your communication peripheral's port setting) leading to the neighbor device.

A "neighbor" is indicated in a *D17 Routing Table by the fact that there is a single PakBus Address present before the hop metric response time. For example, if device 0003 is a neighbor to a datalogger, the datalogger's routing table will look like this:

01: 0003   PakBus Address of destination node
01: 1.0000 worst case response time in seconds

Although *D17 shows neighbors, in the case of multiple peripherals it does not show the ports used to access the respective neighbors. The port used is a function of the P190, neighbor filter, etc. used to discover the neighbor.

7.4 Discovery and Removal of Neighbors

Neighbor filter hello-ing, beaconing or any form of directed communication (such as P190) can be used to discover a datalogger's neighbor. This is followed by a hello-exchange to pass communications verification intervals so that any unresponsive neighbor can be deleted. The *D18 beacon is the simpler of the two methods to configure. More setup is required for a *D19 neighbor

filter which must describe all potential neighbors, but beacon traffic can be reduced or eliminated so long as neighbor communications are usually successful.

The P193 instruction does not initiate communications so it cannot be relied upon to discover neighbors. The P190, P191, P192, P196, P198 and P224 instructions do initiate communications, so if a node is not already a neighbor by another means they will use the static route specified in the Pxxx instruction parameter (port and address) to try to establish that node as a neighbor.

As the basic mechanism of discovery in a PakBus network, whenever a device's neighbor list changes, whether the neighbor is lost or gained, it sends its new neighbor list to all network routers (after a random delay of 1 to 32 seconds). Whether a PakBus device loses or gains a neighbor it will send around its neighbor list. If a neighbor router fails to respond to a Send Neighbor List packet (even if the neighbor router happens to be LoggerNet) it will be removed from the device's neighbor list after 4 attempts.

When a device sends a hello message, either in response to a beacon or because of a neighbor filter, it also sends its communications verification interval to allow its possible future removal as a neighbor.

7.4.1 Beacon

NOTE For RF400 networks of more than a few nodes, beacons should be avoided as the rf collisions and resultant retries can increase network response times to unacceptable values.

Beaconing is a simple way for a device to discover its neighbors. When one device hears another device beaconing, if the beaconing device is not yet a neighbor it will answer the beacon with a hello message back to the beaconing device. The beaconing device will answer with a hello response. The 'hello exchange' will establish the two nodes as neighbors. The CR205 response to a beacon is somewhat different (see 7.7.2).

Most PakBus devices are capable of beaconing (except the CR205). When configured for beaconing a device sends out a broadcast beacon packet at regular intervals. For LoggerNet the default beaconing interval is 60 seconds. A setting of zero disables beacons. The beacon interval ranges are: 0 to 9999 seconds for a datalogger; 0 to 999 seconds for the LoggerNet Server; and 0 to 65535 seconds for the NL100.

An example broadcast beacon in LoggerNet's Low Level Log:

```
09:24:38.558 T bd bd bd bd bd bd bd 8f ff 8f fe 0f ff 0f fe e6 60 bd
```

"8f ff" is the indicator of a broadcast beacon. The two bytes following the "8f ff" contain the beaconing device's PakBus Address in hexadecimal form. In this example "8f fe" indicates that the LoggerNet server (4094 = ffeh) is the beaconing device. The "bd"s are sent for baud rate detection if not connected (datalogger offline).

Once nodes establish themselves as neighbors, if a node does not hear from the other (either by beacon or by addressed communication) within the communications verification interval established in the hello exchange × 2.5, then that node will issue a hello message (up to 4 times if unsuccessful) and delete that node from its neighbor list if the hello message is not answered.

A node will ignore a certain neighbor device's beacons and only listen to addressed communications from that device if the node's own beacon interval is more than 3 times the neighbor device's beacon interval.

A node always responds to a *directed* packet, even though it may not come from a neighbor.

If the beaconing device is a potential neighbor of a certain device with neighbor filter, that device will respond.

A non-beaconing node can initiate a hello message to a beaconing node unless the non-beaconing node has a Neighbor Filter (and the beaconing device has the normal PakBus Address < 4000). This is so that a beaconing node can be sure to discover a non-beaconing node.

Beacon Interval

You configure LoggerNet's beacon interval by clicking on "PakBusPort" in the Setup device map. To set a datalogger's beacon interval you use *D18 with the keyboard/display or Edlog's Options, PakBus Settings. If the datalogger is already programmed for beaconing you will see a non-zero value in *D18 window 01, 02, 03, or 04. The window to which you input the beacon interval selects either a CSDC 7, CSDC 8, Modem Enabled, or RS-232 port for the beacon. You program an NL100's beacon interval in NL100 Edit mode using Telnet (see NL100 manual for details).

When doing scheduled collections, the rule-of-thumb is to make the beacon interval less than the scheduled collection interval. However, to minimize beacon caused delays in the middle of large data transfers, the LoggerNet beacon interval should be greater than the time it takes LoggerNet to collect each scheduled collection. For example, 5 minute scheduled data collection and 1 minute beaconing should work if data collection takes less than 1 minute.

7.4.2 Neighbor Filter

A way to discover neighbors without beaconing is to set up Neighbor Filter *potential neighbors.* Knowing who a datalogger's neighbors are going to be, you can list them. You can include these settings in the datalogger's program using Edlog's PakBus Settings, or directly using *D19. When you set up a neighbor filter the datalogger will send a hello message to each potential neighbor. If the potential neighbor responds it will be put in the datalogger's neighbor list.

The advantage of Neighbor Filter over beaconing is that you avoid generating the periodic rf activity, and the possibility of rf collisions so long as there is some kind of communications within the interval you set, such as a scheduled data collection. More importantly, a neighbor filter avoids selecting marginal rf routes.

The disadvantage of having a Neighbor Filter is that if a potential neighbor fails, the datalogger who lists the failed device as a potential neighbor will continuously send brief hello messages to the failed device at random 1 to 15 second intervals. If this happens, you can use Edlog \ Options \ PakBus Settings to prepare a program to delete the failed device from the datalogger's potential neighbor list until you repair or replace the failed device.

Neighbor Filters provide a way to force a certain route. When beaconing you can't control who answers and what links are established. With Neighbor Filters you can limit what links are established and, thereby, control the routes followed in the network.

If a datalogger has a Neighbor Filter configured, only the devices in the potential neighbor list are candidates for neighbor, with one exception: if a device with Neighbor Filter is addressed (or beaconed) by a node whose address is ≥ 4000, it will treat that node as a potential neighbor. This is so that a datalogger with Neighbor Filter can have LoggerNet, with default PakBus Address of 4094, in its *neighbor* list even though LoggerNet is not in its *potential neighbor* list.

If you have two in-range routers using neighbor filters, in order for them to discover one another you must list each of them as a potential neighbor in the other's neighbor filter. Any device with neighbor filter will ignore hellos unless from a device that is on its list of potential neighbors (or has address ≥ 4000).

If a datalogger has both a Neighbor Filter and a P190, the Neighbor Filter must contain the P190 addressed remote device or the P190 will not be able to communicate with that device. This is by design so that a datalogger with P190 can be Neighbor Filter forced (by leaving the remote device out of the datalogger's Neighbor Filter, but including the desired *router* in the Neighbor Filter) to route through a router rather than directly connect via marginal rf link to a remote device.

If you have previously set up a Router Table in *D15, when you configure Neighbor Filter settings the datalogger sends potential neighbors a hello message. If they respond, the *potential neighbors* become *neighbors* enabling the datalogger to communicate with those devices.

A datalogger's neighbors can be observed by viewing the datalogger's Routing Table. This is accomplished by keying in *D17 and pressing "A" repeatedly to see the table entries. A route consists of 3 lines in the table. If the first two windows of a route contain the same PakBus Address, that device is a neighbor to the datalogger. For example, if device 0003 is a neighbor to the datalogger, the datalogger's routing table will look like this:

01: 0003	PakBus Address of destination node
01: 0003	via neighbor with PakBus Address (this line not present if destination is a neighbor)
01: 1.0000	maximum response time in seconds

7.4.3 Communications Verification Interval (CVI)

A PakBus device with neighbor filter verifies its neighbors by sending each of them a hello message according to the programmed communications verification interval. If a device in its neighbor list does not communicate for a period of time equal to 2.5 × the CVI, it will initiate up to 4 hellos and, if no response, that device is removed from its neighbor list.

A PakBus device passes its CVI during hello exchanges. A beaconing device, such as a datalogger with a *D18 beacon, passes its *beacon interval* as its CVI. A device with neighbor filter simply passes its communications verification interval.

Any device whose beacon interval happens to be set to zero, passes an infinite CVI (LoggerNet, NL100, datalogger). The CR205 passes an infinite CVI in every hello exchange. A non-CRBasic datalogger running a P190, with no neighbor filter nor beacon, also passes an infinite CVI.

7.4.4 Neighbor Removal

7.4.4.1 Automatic

PakBus devices use the communications verification intervals established in their hello exchanges to remove neighbors (after 4 attempts) that have not communicated in some way for a period of time equal to 2.5 × the CVI. To establish these intervals, both nodes in the hello exchange use the longer of the two CVIs.

In the case of a PakBus device having a neighbor filter potential neighbor list, if a neighbor is not heard from (in normal communications) for 2.5 × the neighbor filter CVI, the device will send a hello command to the neighbor. If the neighbor answers with a hello response then the neighbor is retained in the device's neighbor list, otherwise, it is removed. An unresponsive neighbor results in randomly spaced (1 to 15 seconds) hellos as the device attempts up to 4 times to preserve its neighbor. This continues until either the device is removed by the user from the potential neighbor list or until communications is restored.

Whether the CR10XPB, CR510PB or CR23XPB PakBus device is configured as a router or a leaf node, it will always try 4 hellos after the 2.5 × CVI expires before removing a neighbor.

The CR200 Series handle this differently. After the 2.5 × CVI period expires the neighbor is removed without hello attempts.

7.4.4.2 Manual

To manually reset a device's neighbor list, you can send a program to a datalogger with the desired potential neighbor list in the program (see Edlog PakBus Settings section of this guide).

7.4.5 Hello Exchange

A Hello Exchange is the way two devices are established as neighbors.

When a network device beacons, any non-neighbor CRxxxPB device within range responds after a random delay of 1 to 4 seconds by sending a hello command (09) packet to the beaconing device. The beaconing device answers back with a hello response (89). In this manner the beaconing device and the responding devices are established as neighbors. At this point, after a random delay of 1 to 32 seconds, both network routers send their neighbor lists to their neighbor routers allowing all network routers to quickly obtain neighbor lists from every other router.

If a beaconing router has multiple neighbors, it may take more than one beacon to discover them all. The hello responses to a beacon take only 30 milliseconds or so at 9600 baud, but a collision could sometimes occur. If so, the next beacon will probably successfully discover those devices.

If a device hears a (directed) message addressed to it, whose source is not in its neighbor list, it will initiate a hello message to the source device if a neighbor filter does not prevent it.

If LoggerNet is beaconing and the responding device is a CR205, after some low level exchanges LoggerNet sends it a 09 hello command packet.. The CR205 responds with an 89 hello response indicating a 0 (infinite) communications verification interval. This way LoggerNet will keep the CR205 in its neighbor list until LoggerNet's routing table is reset.

A Hello exchange also passes hop metrics, communications verification intervals, and router indication (true/false). After a hello exchange both neighbors use the smaller of the two CVIs that were passed to calculate the 2.5 × CVI to check for neighbor response.

The Hello Exchange is conducted at the highest possible priority so that it will complete ahead of or at the same time as the get and send neighbor list transactions.

Example 1 – LoggerNet (ffe) is beaconing and CR10XPB (001) is in-range.

LoggerNet broadcast beacon:
```
14:17:20.978 T bd bd bd bd bd bd bd 8f ff 8f fe 0f ff 0f fe e6 60 bd
```

CR10XPB sends hello command to LoggerNet:
```
14:17:23.821 R bd af fe f0 01 0f fe 00 01 09 fb 01 01 ff ff 82 7a bd
```

LoggerNet sends hello response to CR10XPB:
```
14:17:23.821 T bd a0 01 3f fe 00 01 0f fe 89 fb 01 03 00 1e 01 4b bd
```

Example 2 – LoggerNet (ffe) is beaconing and CR205 (003) is nearby:

LoggerNet broadcast beacon:
```
15:37:21.665 T bd bd bd bd bd bd bd 8f ff 8f fe 0f ff 0f fe e6 60 bd
```

CR205 responds with hello request (0e) message:
```
15:37:21.800 R bd af fe 00 03 0f fe 00 03 0e e6 60 bd
```

LoggerNet sends CR205 hello (09) message:
```
15:37:22.133 T bd a0 03 7f fe 00 03 0f fe 09 2a 01 03 00 1e f1 5b bd
```

CR205 sends hello response (89) to LoggerNet:
```
15:37:22.368 R bd af fe 00 03 0f fe 00 03 89 2a 00 01 00 1e ff 2b bd
```

7.4.6 Hop Metric

The Hop Metric is intended to be an estimate of the worst case time required to complete a simple transaction. It should include the time it would take to: dial, send a short command, and receive a full length response. It is used by the routing protocol to get an idea of link performance. It is also used to determine network latency between routers for distributing routing information.

During a hello exchange (transaction) both nodes will advertise a Hop Metric that they estimate to be valid for the link. After the exchange both neighbors know how long the other neighbor estimates a worst case exchange to take. The longer estimate of the two is passed on to the routing protocol as the Hop Metric for the link.

Hop metrics determine response time, which is the time that it takes to move a packet from one node to another.

Hop Metrics are necessary to executing the 'shortest path routing algorithm' that every PakBus router uses to choose the shortest (fastest) route to a destination node.

Hop metrics (in seconds) can be viewed in a datalogger's *D17 Routing Table.

7.4.7 Goodbye Message

When a device's PakBus Address is changed (via *D15 or sending a program with PakBus Settings), a goodbye-message using the old address is broadcast out all ports which have a neighbor. This allows the network to update viable links.

Example Goodbye message:
```
bd 8f ff 90 01 0f ff 00 01 0d 00 18 37 bd
```

PakBus device address 001 is broadcasting (to xf ff) the good-bye (od) message.

With an "Is Dialed" setup, LoggerNet sends a Goodbye Command to the device before disconnecting so the device will know not to hello LoggerNet.

7.4.8 Broadcast Packets

The most common use of broadcast packets by a PakBus device is to send beacons for link discovery and verification. The interval at which these messages are sent is controlled by a beacon interval setting that is port specific (the server has a similar setting for its PakBusPort device type). When beacons are sent they are generally sent as an empty message with a high protocol code of type PakCtrl.

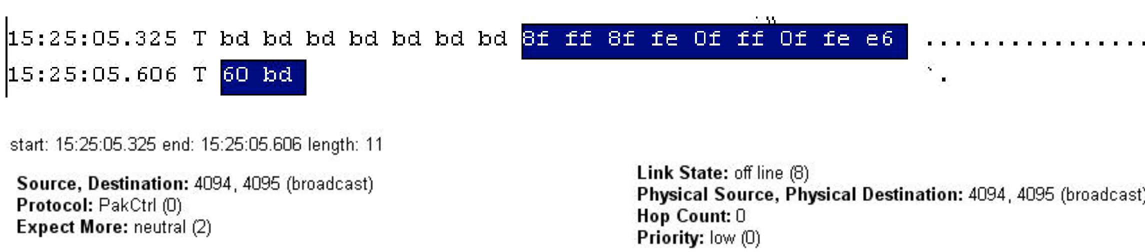
There are other messages that can be sent to the broadcast address:

- When the server is going to hang up the link used by a PakBus port device, it will send a "Goodbye" message with the destination set to the broadcast address. This should have the effect of removing the server from any neighbor's neighbor list. This will prevent the datalogger from attempting to verify the neighbor on a link that is no longer functioning. Dataloggers will also use the goodbye message when their tables are reset for various reasons (such as changing the PakBus address).

- Under client control, the server will send a "Hello" command to the broadcast address. For instance, this is done when the user selects the "Search for Neighbors" option in the right click menu for the server in PakBusGraph. There is a similar function in the setup screen. This is actually an alternative form of beaconing but is one that will more reliably give us a response (if a datalogger thinks it knows us as a neighbor, it won't respond to a conventional beacon message).

- Dataloggers can be set up to send time update messages to a broadcast message. This provides a means of keeping clocks of various devices in close synchronization across a network.

In Log View (with PakBus Filter added) a broadcast packet might look like this:



There are some characteristics that all broadcast messages share:

- Broadcast messages should not expect an immediate response.

- The transaction number either is ignored or is not present in the message.

- A broadcast message will never be forwarded. They will always be exchanged between neighbors.

7.5 Router

A router is a PakBus device which can accept a packet and forward it toward its network destination. Network routing is fundamentally driven by a router learning who its neighbors are and then sharing that information with other routers in the network. Send Neighbor List and Get Neighbor List packets are used to accomplish the sharing.

A router device which has experienced some change to its neighbor list (either a neighbor added or a neighbor deleted) will exchange neighbor lists with other routers so that all the routers in the network have the neighbor lists of all the other routers. For example, after a router participates in a hello exchange it sends all other routers in the network its updated neighbor list.

A router will only route a received packet if the device has the packet's destination PakBus Address in its routing table, however, if the device has a *default router address* configured in the *D15 4th window, and is unable to find a route to the destination in its routing table, then it will send the packet out via the default router. A functioning network device will eventually be known to every router in the network.

The CR10X, CR510, CR23X (with PakBus OS), the NL100, and LoggerNet can function as routers. Typically, network routers consist of one of the above dataloggers with an RF400 Series radio attached. A router's RF400 can communicate both directions (i.e., toward LoggerNet or toward the destination datalogger). The future RF400 Series radio will include the ability to function as a stand-alone router (without a datalogger).

If you have two in-range routers using neighbor filters, in order for them to discover one another you must list each of them as a potential neighbor in the other's neighbor filter.

Routers determine the 'best complete route' to a destination by traversing through all the neighbor lists they maintain from all the routers. These neighbor lists give all routers knowledge of every *link* in the network. To limit size, routers do not maintain in their routing tables 'best complete routes' to every node in the network, although they have enough information to calculate such routes. Routers need only maintain a list of PakBus Addresses in the network and the corresponding neighbor device through which to send a packet toward each PakBus Address.

7.5.1 Routing Table

The Routing Table lists the shortest (fastest) possible route to each node in the network. All routers use the same 'shortest path routing algorithm' to construct their routing tables. Every device that is configured as a router maintains a routing table in memory with a list of active network links. In order to become a router, a datalogger's routing table is configured in *D15 or using Edlog's Options \ PakBus Settings.

The routing tables of a non-CRBasic datalogger (CR10XPB, CR23XPB, and CR510PB) can be viewed by pressing *D17. The NL100's Routing Table can be viewed by using telnet, or a terminal emulator on the RS-232 port to access

its command/setup interface. The "t" or "tables" command will report the NL100's current routing tables.

Every router (including LoggerNet) sends around its neighbor list to every other router. Routers construct their Routing Tables using their own neighbor list and the neighbor lists (all with hop metrics) received from the other routers in the network.

A device will be removed from another device's neighbor list (and routing table) if a hello exchange fails after 4 attempts. The significance of this is, for example, that if a CR10XPB is a router to a CR205, but is not trying to issue a P190, P191, P192, P196, P198 or P224 message, then the CR10XPB can only continue to route packets to it if it maintains the CR205 as a neighbor, or is able to route through a router that has the CR205 as a neighbor.

Reset Router

To reset a datalogger router using a CR10KD, enter *D15 and re-enter the PakBus Address, or if there is a neighbor filter, you can zero the port code and re-enter it.

You can reset a router under program control using P192 with code 12 in Parameter 3.

7.5.2 LoggerNet Routes

LoggerNet has two different routing tables. One is for 'Static' Routes and the other for 'Dynamic' Routes. Either kind of route can be used to establish a connection.

7.5.2.1 Static Routes

Static routes exist by virtue of a device map. They are theoretical routes via a presumed neighbor. LoggerNet falls back on a static route to attempt a connection with a device in the absence of a dynamic route. For example, LoggerNet can use a static route to connect with a device before there has been sufficient time to establish a dynamic route via beaconing, or where LoggerNet beacons have been disabled by making the beacon interval = 0. Once a connection is made, the device becomes a neighbor to LoggerNet and remains so until the LoggerNet router is reset.

7.5.2.2 Dynamic Routes

Dynamic routes are learned routes resulting from communications which triggered a hello-exchange.

Dynamic routes are routes learned in the standard PakBus ways. If there is no dynamic route, then LoggerNet uses the "static" route, which can either succeed or fail. If it succeeds, then, normally, the PakBus discovery mechanism takes place (hello-exchange between neighbors, passing neighbor lists among routers) and a dynamic route is created.

7.5.3 Forced Routing

7.5.3.1 Tree vs. Flat Device Map

If you build the device map with the remote PakBus devices as children of the 'base' router you have a 'flat' device map configuration. Un-represented RF400 Series are assumed here with all hop sequences, network addresses, radio addresses, and standby modes (normally "< 24 mA always on) configured the same.

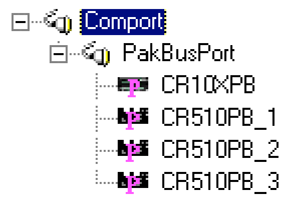


FIGURE 19. Flat Map

If you build the map with succeeding remote devices as children of the previous device, it is a 'tree' configuration (un-represented RF400 Series radios are assumed).

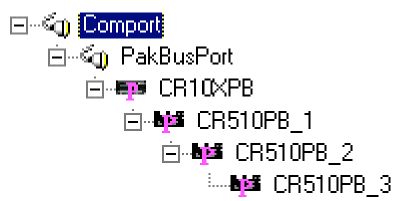


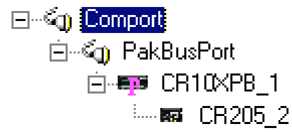
FIGURE 20. Tree Map

Which configuration you choose (flat or tree) makes a difference in terms of the LoggerNet static routes produced. When a tree configuration exists and a static route is being used, LoggerNet will route through a parent device en route to a child device. Static routes can only govern the first hop, however, so the difference between the flat and tree maps is small.

Once a dynamic route has been established, LoggerNet uses it in preference to a static route.

7.5.3.2 Forced Routes

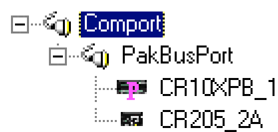
To force a route you disable LoggerNet beacons and beacons from other devices and configure router devices with neighbor filter potential neighbors. In the absence of beacons, LoggerNet will use the device map (static routes) in selecting a first hop neighbor, then calculate the fastest route via neighbor lists in network devices' router tables to route a packet to its destination. Take, for example, the following device map (un-represented RF400 Series are assumed):



If there is no beaconing in the network, LoggerNet can connect to CR10XPB_1 because it is in LoggerNet's static routes. If CR10XPB_1 does not have CR205_2 in its neighbor list, LoggerNet's routing algorithm cannot route toward CR205_2. By adding CR205_2 to CR10XPB_1's neighbor filter potential neighbor list, you essentially force that route to CR205_2.

If LoggerNet were beaconing, it could discover CR205_2 directly (sometimes creating an undesirable, weak rf link).

If you had the following device map (un-represented RF400 Series are assumed):



With LoggerNet not beaconing, you can connect to CR205_2 via a static route, only it will not be via CR10XPB but directly to CR205_2A.

Neighbor Filter Force A Path Through Router

If a datalogger has a *D19 neighbor filter, only the devices in its potential neighbor list are candidates for neighbor. An exception is that if the datalogger hears a PakBus node with address ≥ 4000 (normally the LoggerNet server with PakBus Address = 4094) it will send a hello message to it and they will become neighbors.

For example, if a datalogger has both a neighbor filter and a P190, the neighbor filter must list the P190 addressed remote device or else the P190 will not be able to communicate with the remote device. This is by design so that a datalogger with P190 can be forced (by leaving the remote device out of the P190 datalogger's neighbor filter but including the desired *router*) to take the path through the router avoiding the undesirable long path (marginal rf link) to a remote device (see Figure 21).

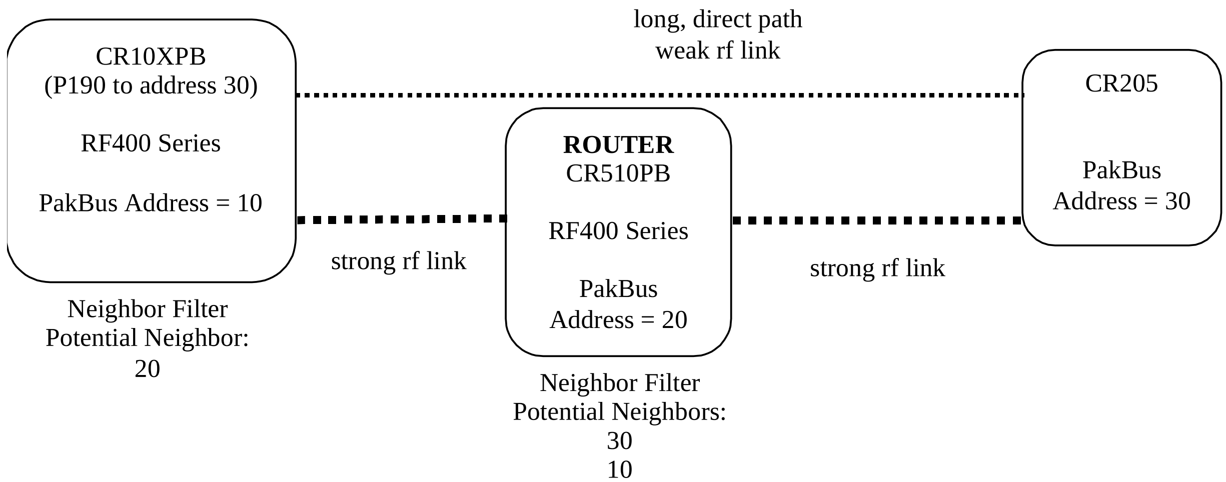


FIGURE 21. Neighbor Filter Force a Path through Router

7.6 Concurrent Communications

PakBus OSs allow dataloggers to simultaneously communicate with multiple peripherals using interleaved packet transfers. With an SDC peripheral, packet arbitration takes advantage of the synchronous device's *addressed enabling*. In the case of a *modem enabled* peripheral, the datalogger uses a PakCtrl level "pause"/"finished" exchange to momentarily pause the M.E. peripheral's link while another (non-M.E.) peripheral transfers a packet.

As with mixed-array OSs, only one M.E. peripheral can be connected to a PakBus datalogger. Several SDC peripherals can be connected and communicating at once. For example, a CR10XPB can connect to: an SC32B (M.E.), an RF400 (CSDC 7), an MD485 (CSDC 8) and the CR10KD (SDC 6). They can all be communicating at the same time.

Dual LoggerNet Servers

It is possible to configure two servers in a network, each displaying real time data on their Numeric Displays. They roughly alternate receiving records from the datalogger. One server can ring up the datalogger while another sever is already connected. The key to getting the second server set up is to make sure that in Options \ LoggerNet Settings (Toolbar) its PakBus Computer ID is set to a different ID than the first server. To take effect, the ID must be changed *before* building the device map. In order to monitor Input Locations in Numeric Display, once you connect with the second server, it is necessary to Get Table Definitions by highlighting the datalogger in Setup, clicking on the Data Files tab, and clicking the Get Table Definitions button.

7.7 CR200 Series

7.7.1 Beaconing, Neighbor List and Comms Verification Interval

In this section the term "CR205" includes the CR215, both having internal radios, and the CR200 in some networks. The CR205's neighbor 'list' will accept only one device. If a CR205 has no device in its neighbor list it will respond to a beacon after a random 0 to 1 second delay by sending a *hello-request* message to the beaconer. The beaconing device (if unaware of the CR205) answers by sending a hello message addressed to it. The CR205 sends back a hello message response indicating the same communications verification interval as the beaconer and adopts the communications verification interval as its own.

After a CR205 responds to a beacon and the hello-exchange is complete, the beaconing device tells the CR205 to send its neighbor list. The CR205 complies and they are established as neighbors.

If 2.5 × the communications verification interval passes without hearing from its neighbor, the CR205 resets its neighbor list and becomes open to acquiring a neighbor once more.

A CR205 always responds to a *directed* packet, even though the packet may not come from a neighbor.

7.7.2 Routing

A CR200 Series datalogger cannot be a router. In a PakBus network a CR200 Series is always a 'leaf node.'

8. Troubleshooting

8.1 LoggerNet can't communicate with in-range PakBus datalogger

(PC-RF400~~~RF400-CR510PB~~~ RF400-CR510PB~~~CR205)

Possible reason 1
LoggerNet's PakBus Address for datalogger doesn't match datalogger's PakBus Address.

Remedy 1
Make them match.

Possible reason 2
An RF400 Series radio is set to a different Hopping Sequence, Network Address, Radio Address, or Standby Mode.

Remedy 2
Set both radios exactly the same in the above parameters.

('base' radio's Active Interface is typically Auto Sense, remote radio is typically CSDC 7)

Possible reason 3
No PakBusPort in device map between root and datalogger.

Remedy 3
Add PakBusPort.

Possible reason 4
No PakBus OS in datalogger

Remedy 4
If you have two in-range routers using neighbor filters, in order for them to discover one another you must list each of them as a potential neighbor in the other's neighbor filter.

8.2 LoggerNet can't communicate via datalogger-router to a certain remote datalogger.

(PC-RF400~~~RF400-CR10XPB~~~RF400-CR510~~~CR205)

Possible reason 1
A datalogger router has insufficient *D15 max nodes, max neighbors, or max routers configured.

Remedy 1
Increase the *D15 numbers.

*D15 settings of 000x, 6, 6, 6, 0 are reasonable for a router in a network of under a half dozen nodes (see *D15 Set PakBus Address and Routing Table, Section 4.1.2.1).

CAUTION *D15 settings allocate memory similar to *A. It is a good idea, when configuring *D15 settings, to leave 'room to grow.'

Changing *D15 settings later on could result in loss of data as *0 is entered to compile new settings.

Possible reason 2
Last datalogger router has no means configured of discovering the remote datalogger.

Remedy 2
Configure the datalogger router with either a *neighbor filter* or a *beacon*. Make sure Neighbor Filter potential neighbors include the remote datalogger's address. Whether you set up a beacon or Neighbor Filter make sure the port so configured matches the communications device port configuration. For example, if the selected neighbor filter port is "17", make sure that the RF400 Active Interface is "CSDC 7."

Possible reason 3
An RF400 Series radio is set to a different Hopping Sequence, Network Address, Radio Address, or Standby Mode.

Remedy 3
Set all network radios exactly the same in the above parameters. Network RF400s' Active Interfaces may vary from node to node, however, they will typically be configured for CSDC 7 or 8 except for a 'base' radio which is typically AutoSense or M.E.. Dataloggers automatically detect the RF400's port (Active Interface) for packet communications, however, the potential neighbor hello port or beacon port must be configured to match the RF400's Active Interface, or no discovery of neighbors will take place.

Possible reason 4
A CR200 Series has just received an OS download resetting network address, radio address, hopping sequence, and radio power mode to defaults.

Remedy 4
Configure CR205 settings to agree with network.

Possible reason 5
The two routers in the path to the CR205 have neighbor filters and at least one of the neighbor filters doesn't list the other as a potential neighbor.

Remedy 5
If you have two in-range routers using neighbor filters, in order for them to discover one another you must list each of them as a potential neighbor in the other's neighbor filter.

8.3 Can't LoggerNet communicate with datalogger-router in network?

(PC-RF400~~~RF400-CR10XPB~~~CR205)

Possible reason
The datalogger-router has more than one M.E. peripheral cabled to it.

Remedy
Connect only one M.E. device to a datalogger (or change one M.E. peripheral to CSDC 8 port).

8.4 Changed P190 port type and it no longer communicates with remote.

Possible reason
The Active Interface of the communications device (for example, RF400 Series) no longer matches the P190 port.

Remedy
Make the communications device Active Interface agree with P190 Parameter 1.

8.5 Rapid spurious communications lasting a few seconds at a time between devices in RF400 network.

Possible reason
Two network devices have the same PakBus Address.

Remedy
Change one of the duplicate PakBus Addresses. Make all addresses unique throughout the network.

8.6 In P193 network, certain CR200 Series devices don't transfer data.

Possible reason
The Master datalogger's *D15 setting configures too few max nodes, max neighbors, and max routers.

Remedy
Change Master *D15 settings for max nodes, max neighbors, and max routers to larger numbers reflecting the network size.

CAUTION *D15 settings allocate memory similar to *A. It is a good idea, when configuring *D15 settings, to leave 'room to grow.'

Changing *D15 settings later on could result in loss of data.

Appendix A. Glossary of PakBus Terms

Active Interface An RF400 Series' or MD485's selected com port. It could be SDC, CSDC 7, CSDC 8, Modem Enabled, or RS-232.

Beacon A broadcast packet intended to discover in-range neighbor PakBus devices. Devices that receive a beacon answer by initiating a hello-exchange with the beaconing device if not already neighbors. A device with neighbor filter will ignore beacons from addresses not in their potential neighbor list unless the beacon address is ≥ 4000.

Beacon Interval User assigned interval between broadcast beacons.

BMP5 Block mode protocol 5. The datalogger application layer packet protocol used in PakBus communications.

Broadcast Sent to all PakBus devices in the network within range.

Non-CRBasic PakBus Datalogger CR10X, CR510, or CR23X datalogger.

Client The client part of the client-server architecture. Typically, a client is an application that runs on a personal computer that relies on a server to perform certain operations. A PakBus example: ConnectScreen and Setup in LoggerNet are clients to the LoggerNet server.

Communications Verification Interval The interval of time that a PakBus device uses to determine when it's time to send a hello message to a neighbor to see if it can still respond and remain a neighbor. The CVI is normally multiplied by a factor of 2.5 for such use. CVIs are passed in hello exchanges.

Concurrent Communications The ability of a PakBus datalogger to communicate with several peripherals at the same time.

CSDC Concurrent Synchronous Device Communications. Clocked communications between datalogger and addressed peripheral which can be interleaved with communications from other addressed peripherals.

CS I/O Campbell Scientific I/O interface. An interface with 9-pin connector which has the appearance of a 9-pin RS-232 interface but different line assignments and functions.

CVI	Abbreviation for Communications Verification Interval.
Directed Packet	A relative term; from the vantagepoint of a particular PakBus device, a directed packet is a packet that is addressed to the particular PakBus device. It is also referred to as "addressed communications."
Dynamic Link	A learned route, distinguished by the fact that the PakBus device addresses exist in one another's neighbor lists.
Edlog	The software editor for non-CRBasic datalogger program creation and edition. Supports mixed array, Table Data, and PakBus programs.
Expect More	A low-level indication in the packet header of whether or not the function being accomplished by the packet is finished or will involve more packets to come.
Flat Map	LoggerNet Setup device map construction where all PakBus devices are children of the PakBusPort (instead of coming off the device above).
Forced Route	A route that is determined entirely by Neighbor Filter potential neighbors instead of beaconing.
Goodbye Message	The message broadcast by a PakBus device if its PakBus address changes. A Goodbye Message is sent out all ports having a neighbor, enabling network routers to learn the change.
Header	The leading part of a PakBus packet containing such information as: source, destination, link status, etc.. See PakBus Concepts section.
Hello Exchange	The sending out of a hello command packet by one PakBus device to another device, and the receiving of the hello response to that packet from the neighbor. Only a hello exchange can establish two devices as neighbors.
Hello Message	A discovery packet sent to a potential neighbor or sent in response to a beacon. Sending a hello message is sometimes referred to as "hello-ing."
Hop	A link in a communication path. Communication with a neighbor.
Hop Metric	A measure issuing from a hello exchange indicating the maximum response time to obtain a response from the other node.

Hopping Sequence Also called Hop Sequence or Hop Table, this refers to a particular pattern the spread spectrum radio uses to 'hop' among its assigned frequencies. In a PakBus network all RF400s must have the same hopping sequence.

Leaf Node A PakBus device which cannot serve as a router, although it can send or receive packets.

Link A viable communication path between two PakBus devices.

Link State Network level: offline, ring, ready, finished, or pause status.

LogView Stand-alone software for decoding and viewing PakBus packets from a low-level log file.

M.E. Modem Enabled.

Modem Enabled Datalogger peripherals that are enabled by the CS I/O modem enabled line (line 5) going high.

Neighbor From the perspective of a particular PakBus device, a neighbor is a device with which it has recently communicated directly (not via router).

Neighbor List List of neighbor(s) that every PakBus device maintains. Leaf nodes can only have a single neighbor. Router nodes can have many neighbors. Router devices pass their neighbor lists to other routers in the network to build the network routing system.

Neighbor Filter The preferred means of discovery for rf networks. A group of datalogger settings that controls which PakBus devices a datalogger may attempt to establish as neighbors. A neighbor filter initiates hello exchanges from its *potential neighbor* list, ignoring (filtering out) any beacons from non-potential neighbor sources unless their PakBus Address is ≥ 4000 (normally LoggerNet). If not satisfied by normal communications, the neighbor filter verifies communications with established neighbors by initiating hello exchanges according to communications verification intervals.

Network Address RF400 network address. In a PakBus network all RF400s must have the same network address.

Node Another term for a PakBus device in a network.

Packet A frame, typically 1000 bytes or less, containing a header and data that is transferred over a packet-switching network. Packets transfer information between LoggerNet and PakBus dataloggers usually via routers.

P19x PakBus instructions for non-CRBasic PakBus capable dataloggers such as the CR10X. For example, the P193 instruction is used for master/slave communications with a group of remote CR200 Series wireless sensors. This group includes the P224 instruction.

PakBus Address Unique address assigned to a PakBus device in a PakBus network.

PakBus Device A device with processor and software to handle PakBus packet communications (BMP5).

PakBusGraph LoggerNet client that graphically depicts PakBus devices in the network and allows editing and/or viewing of device settings such as routing table, port settings, protocol capabilities, etc.. Essentially, PakBusGraph displays LoggerNet's routing table.

PakCom Software for network setup of RF400 Series and CR200 Series devices.

PC Address LoggerNet's PakBus address.

PakCtrl Network level PakBus communications protocol. For example, PakCtrl packets perform discovery and routing functions such as doing hello exchanges and sending neighbor lists.

PakSer NL100 configuration where PakBus packets are converted to serial data for non PakBus dataloggers.

Physical Destination The neighbor device who received the packet.

Physical Source The neighbor device who sent the packet.

Ping Solicit response from network device. Send it a packet and receive packet back in response.

Port Com port. Datalogger or peripheral interface: SDC, CSDC, Modem Enabled, or RS-232.

Potential Neighbor A PakBus device specified in a neighbor filter which the router is allowed to 'hello' and attempt to establish as a neighbor.

Potential Neighbor List — A list of possible neighbors installed in a datalogger's neighbor filter using *D19, PakBusGraph \ Show Settings, or Edlog \ Options \ PakBus Settings.

Protocol — An agreed-upon format for transmitting data between two devices. The protocol determines the following: any error checking to be used, how the sending device will indicate that it has finished sending a message, how the receiving device will indicate that it has received a message, etc.

Priority — Indicator of packet importance from 0 to 3.

Radio Address — RF400 Radio Address. In a PakBus network all RF400s have the same radio address.

Represented — Device that is shown in the LoggerNet device map. Some communication devices, such as the RF400 Series, are transparent and need not be shown in the device map.

rf — Radio frequency. Having to do with radio transmitters and receivers.

Router — A PakBus device that is configured to accept packets destined for another device and pass them along toward that device. A router only accepts packets destined for PakBus addresses for which it has a route. A router could be a CR10X, NL100 or even LoggerNet (a CR200 Series device can't route).

Routing Table — A router's list of routes (neighbors leading to destinations) to every PakBus device in the network. Created using the shortest link algorithm using the neighbor lists passed among network routers.

SDC — Synchronous Device Communications - clocked serial communications between datalogger and an addressed peripheral. A datalogger peripheral that is enabled by addressing rather than by asserting the modem enabled line.

Server — The server part of the client-server architecture. A computer or device on a network that manages (gives access to) network resources. In Pakbus networking LoggerNet's "cora" server has a router or routers and communicates with other PakBus devices via its PakBus Interface. It communicates using transactions with clients such as Status Monitor, PakBus devices, and ConnectScreen via its LoggerNet Interface.

Standby Mode	RF400's or CR200 Series' low power budget mode. Adjustable in 7 ranges from 'always on' to listening briefly once every 8 seconds (0 to 7). The RF400 default is "<4 mA ½ sec cycle."
Static Route	Route that exists by virtue of a LoggerNet Setup device map or a P19x instruction.
Switched Packet Network	A network where packets are transferred, not by transparent links, but by intermediate routers.
Table Definitions	Information that LoggerNet needs to know about the data stored in a datalogger in order to collect data for real time and historical purposes.
Transaction	An exchange of packets between nodes. An example is the Hello Transaction (Exchange). Most commands expect a response.
Tree Map	LoggerNet Setup device map construction where succeeding PakBus devices are added as children of the device above.

Appendix B. NL100 TcpServer to RF400 to CR510PB-Router/RF400 to 20 CR205s Network

LoggerNet does 60 minute scheduled collections from remote CR205s via NL100 configured as TcpServer and a CR510PB router using RF400 for communications. The CR510 router extends the range from NL100 to CR205s. Additional CR510PB/RF400 'repeaters'/routers can be added to extend any branch in the network, for example, to get over a hill to some of the CR205s.

The NL100 is transparent in this mode (it has no PakBus address). There are no nodes beaconing in this configuration. LoggerNet needs no beacon as it can discover the CR510PB using the 'static route' from its device map. When LoggerNet is first run, having no beacon configured it will be unaware of the CR510PB. But when it first connects (manually or scheduled) with the CR510PB it exchanges neighbor lists with it so LoggerNet can thereafter route to any node in the network. LoggerNet will keep the CR510PB as a neighbor until LoggerNet is closed or until you reset LoggerNet's router in PakBusGraph.

The CR510PB is configured with a Neighbor Filter to discover the CR205s. Also to set up a CSDC 7 port for interface to its RF400. If the CR510PB has had time to discover its *D19 potential neighbors, the CR205s, they appear in PakBusGraph as soon as the LoggerNet to CR510PB connection is made.

The CR510PB can also do measurement/control.

<u>Hardware Setup</u>   ROUTER

PC---(LAN)---NL100$_{(RS232)}$-------$_{(RS232)}$RF400~~~~~RF400$_{(CSDC7)}$--
CR510~~~~~~~~~~~~~CR205_1

~~~~CR205_2
~~~~CR205_3
…

<u>LoggerNet Device Map</u>

IPPort
   PakBusPort
      CR510TD-PB   (PakBus Address = 100)
         CR200_1 (PakBus Address = 1)
         CR200_2 (PakBus Address = 2)
         CR200_3 (PakBus Address = 3)
         …   (PakBus Address = n)

In the PakBusPort screen configure beaconing for 0 seconds. LoggerNet will find the CR510PB as a static route from its device map. The first time you connect, PakBusGraph will show the CR510PB and all the CR205 neighbors to the CR510PB.

LoggerNet Schedule Tab

Collection Interval – 60 minutes
Primary Retry Intervals – 20 minutes
Number of Primary Retries – 2
Secondary Retry Interval – 240 minutes

NL100 Configuration Menu Settings

```
NL100/105:

TLink config: [disabled]

RS485 config: [disabled]

CS I/O config: [disabled]

RS232 config: [TcpSer]
  RS232 bps: [9600]
  RS232 serial server port number: [3002]

EtherNet 10BASE-T: [enabled]
  10BASE-T port IP address: [192.168.8.142]
  10BASE-T port network mask: [255.255.0.0]
  IP address of the default gateway: [0.0.0.0]

PakBus Address of the NL100/105: [55]
  Clock neighbor node Id: [0]

PakBus/TCP server config: [disabled]

PakBus/TCP client config: [disabled]

Modbus/TCP gateway config: [disabled]

Telnet session password: [nl100]

NL100/105 (ver, show, edit, defaults, reset, help, bye):
```

(Any NL100 options not shown above are disabled).

A PakBus Address is required by setup even though not used.

RF400 Settings

In a PakBus rf network all radios must have the same radio and network addresses and hopping sequence setting. Addressing is accomplished using the unique PakBus addresses assigned to the PakBus devices (dataloggers).

1. Configure both RF400 radios with the same Radio Address, Net Address, and Hopping Sequence. If there will be no other RF400s within range, you can use default settings.

2. RF400 connected to NL100

   Active Interface = “AutoSense”
   Standby Mode = “< 24 mA always on”

3. RF400 connected to CR510

   Active Interface = CSDC 7 (7 is default but you could configure 8)
   Standby Mode = < 24 mA

CR205 Settings

1. Configure all CR205s with the same Radio Address, Net Address, and Hopping Sequence using the same settings as the RF400s have.

2. Configure each CR205 with a unique PakBus address ("Datalogger Address" in PakCom) starting with "1" and ending with "20."

3. Configure Radio Power Mode = RF_ON

4. Configure Radio Force On = 0

CR510 (Router) Settings

Configure the Network settings, and Neighbor Filter in the CR510PB program using Edlog\Options\PakBus Settings. When the program is sent to the CR510PB, the program will configure the PakBus settings in the CR510PB.

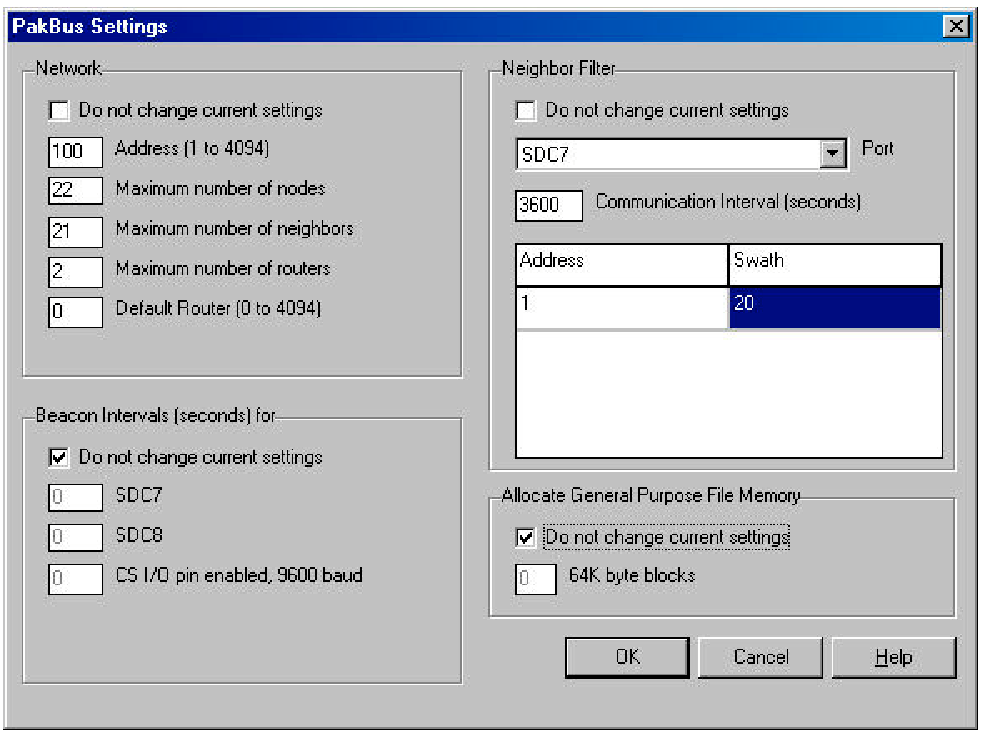


FIGURE B-1. Edlog CR510 Setup

The Communication Interval could be set ≤ the scheduled collection interval. If the CR510PB does not hear from a CR205 within 2.5 × this interval, it will drop the CR205 from its Neighbor List and send hello messages randomly every 1 to 15 seconds until communications is restored or until you remove the CR205's address from the CR510's potential neighbor list. You could use PakBusGraph for this.

Be sure to OK the PakBus settings and save/compile the program in Edlog before sending it to the CR510PB. When you send the program to the CR510PB the above settings should be visible in *D15 and *D19.

After the CR205 has had a few minutes to discover its CR205 neighbors, you can view them in the *D17 Routing Table. The server should also show up as a neighbor to the CR510PB.

Appendix C. NL100 TcpServer to RF400-Router to 20 CR205s Network

In this configuration LoggerNet does 60 minute scheduled collections from remote CR205s via NL100 configured as TcpServer and an RF400 for communications. In this configuration, the CR205s have to be relatively close to the NL100. The network configuration described in Appendix B shows how to extend range with a ‘repeater’/router.

Notice that the LoggerNet, NL100, RF400 and CR205 setups are the same as in Appendix B. The difference is the router which is missing. This demonstrates the fact that adding a router is transparent to LoggerNet. Adding a router only involves configuring the router’s port (CSDC 7, CSDC 8, or Modem Enabled) and configuring the way the datalogger/router will discover its neighbors (beacons or potential neighbors) and you are done.

<u>Hardware Setup</u>

```
PC----(LAN)----NL100(RS232)----(RS232)RF400~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~CR205_1
                                                                      ~~~~CR205_2
                                                                      ~~~~CR205_3
                                                                           ...
```

<u>LoggerNet Device Map</u>

```
IPPort
   PakBusPort  (
      CR200_1     (PakBus Address = 1)
      CR200_2     (PakBus Address = 2)
      CR200_3     (PakBus Address = 3)
        ...       (PakBus Address = n)
```

In the PakBusPort screen configure beaconing for 0 seconds. LoggerNet will find the CR510PB as a static route from its device map. The first time you connect, PakBusGraph will show the CR510PB and all the CR205s which are neighbors to the CR510PB.

<u>LoggerNet Schedule Tab</u>

Collection Interval – 60 minutes
Primary Retry Intervals – 20 minutes
Number of Primary Retries – 2
Secondary Retry Interval – 240 minutes

NL100 Configuration Menu Settings

```
NL100/105:

TLink config: [disabled]

RS485 config: [disabled]

CS I/O config: [disabled]

RS232 config: [TcpSer]
  RS232 bps: [9600]
  RS232 serial server port number: [3002]

EtherNet 10BASE-T: [enabled]
  10BASE-T port IP address: [192.168.8.142]
  10BASE-T port network mask: [255.255.0.0]
  IP address of the default gateway: [0.0.0.0]

PakBus Address of the NL100/105: [55]
  Clock neighbor node Id: [0]

PakBus/TCP server config: [disabled]

PakBus/TCP client config: [disabled]

Modbus/TCP gateway config: [disabled]

Telnet session password: [nl100]

NL100/105 (ver, show, edit, defaults, reset, help, bye):
```

(Any NL100 options not shown above are disabled).

A PakBus Address is required by setup even though not used.

RF400 Settings

In a PakBus rf network all radios must have the same radio and network addresses and hopping sequence setting. Addressing is accomplished using the unique PakBus addresses assigned to the PakBus devices (dataloggers).

1. Configure RF400 radio with the same Radio Address, Net Address, and Hopping Sequence as the CR205s. If there will be no other RF400s in the area, you can use default settings, otherwise choose some random settings and configure all alike.

2. RF400 connected to NL100

   Active Interface = “AutoSense”
   Standby Mode = “< 24 mA always on”

3. RF400 connected to CR510

   Active Interface = CSDC (7 by default but could be 8)
   Standby Mode = < 24 mA always on

CR205 Settings

1. Configure all CR205s with the same Radio Address, Net Address, and Hopping Sequence using the same settings as the RF400 has.

2. Configure each CR205 with a unique PakBus address (“Datalogger Address” in PakCom) starting with “1” and ending with “20.”

3. Configure Radio Power Mode = RF_ON

4. Configure Radio Force On = 0

Appendix D. COM210 to CR510PB-Router to 10 CR205s Network

In this configuration LoggerNet does 15 minute scheduled collections from remote CR205s via COM210 dialed link and a CR510PB configured as router.

Hardware Setup

```
PC----(PC modem)----------------------COM210--CR510PB--RF400~~~~~~~~~~~CR205_1
                                                                  ~~~~CR205_2
                                                                  ~~~~CR205_3
                                                                     …
```

LoggerNet Device Map

```
ComPort_1
   PhoneBase
      PhoneRemote
         PakBusPort
            CR510PB (PakBus Address = 100)
               CR205_1    (PakBus Address = 1)
               CR205_2    (PakBus Address = 2)
               CR205_3    (PakBus Address = 3)
                  …       (PakBus Address = n)
```

In the PakBusPort screen configure beaconing for 0 seconds. LoggerNet will find the CR510PB as a static route from its device map. The first time you dial and connect, PakBusGraph will show the CR510PB and the CR205 neighbors to the CR510PB.

LoggerNet Schedule Tab

Collection Interval – 15 minutes
Primary Retry Intervals – 5 minutes
Number of Primary Retries – 2
Secondary Retry Interval – 60 minutes

RF400 Settings

In a PakBus rf network all radios are configured for the same radio address, network address, and hopping sequence. Device selection is accomplished using the unique PakBus addresses assigned to the PakBus devices (dataloggers).

1. Configure RF400 with the same Radio Address, Net Address, and Hopping Sequence as the CR205s have. If there will be no other RF400 radios within range, you can use default settings.

2. Other settings

   Active Interface = “CSDC” (7 by default but could be 8)
   Standby Mode = “< 24 mA always on”

CR205 Settings

1. Configure all CR205s with the same Radio Address, Net Address, and Hopping Sequence using the same settings as the RF400 has.

2. Configure each CR205 with a unique PakBus address ("Datalogger Address" in PakCom) starting with "1" and ending with "10."

3. Configure Radio Power Mode = RF_ON and Radio Force On = 0

Appendix E. CR510PB w/ P190s to CR205s Network

In this configuration LoggerNet does 15 minute scheduled collections of a remote CR510XPB running P190s to transfer variable values from 5 remote CR205s. This approach results in the fastest data transfer compared to P193 and LoggerNet direct collection of CR205s.

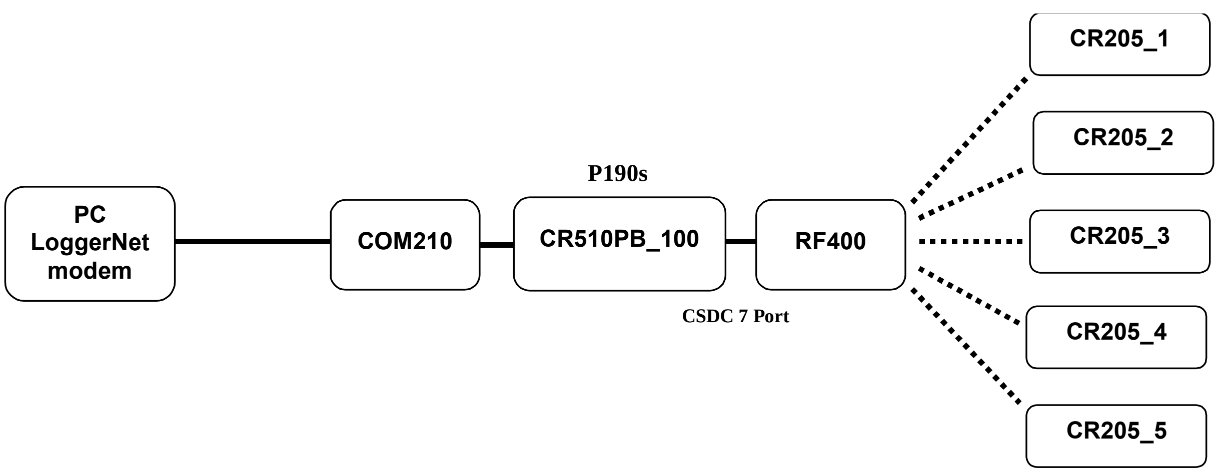
The P190 instruction which has the CR205's address does device discovery in this configuration. One P190 is required for each remote.

The higher data throughput of the P190 configuration is due to the fact that one P190 will continue to the next P190 as soon as it gets a response from the remote. The asynchronous P190 transmissions require the CR205s to always have their radios on resulting in higher remote power budgets compared to the P193 approach.

There are no nodes beaconing in this configuration. LoggerNet needs no beacon as it can discover the CR510PB using the static route from its device map. When LoggerNet is first run, having no beacon configured it will be unaware of the CR510PB. But when they first connect they will do a hello exchange, swap neighbor lists, and become neighbors. LoggerNet can thereafter communicate with any node in the network. LoggerNet will keep the CR510PB as a neighbor until LoggerNet is closed or its router reset.

Monitoring of a CR205's received signal strength is shown.

Hardware Setup



LoggerNet Device Map

ComPort_1
PhoneBase
PhoneRemote
PakBusPort
CR510PB (PakBus Address = 100)
CR200_1 (PakBus Address = 1)
CR200_2 (PakBus Address = 2)
CR200_3 (PakBus Address = 3)
CR200_4 (PakBus Address = 4)
CR200_5 (PakBus Address = 5)

After once dialing up and connecting in LoggerNet, PakBusGraph will display the CR510PB and all its CR205 neighbors.

LoggerNet Schedule Tab

Collection Interval – 15 minutes
Primary Retry Intervals – 5 minutes
Number of Primary Retries – 2
Secondary Retry Interval – 60 minutes

RF400 Settings

In a PakBus rf network, all radios are set up with the exact same radio and network address and hopping sequence.

1. Configure RF400 radio with the same Radio Address, Net Address, and Hopping Sequence as the CR205s. If there will be no other RF400s in the area, you can use default settings, otherwise choose some other settings and configure all alike.

2. Other Settings

   Active Interface = “CSDC 7” (7 is default. It could be changed to 8 if no other peripheral uses 8).

   Standby Mode = “< 24 mA always on” (other modes work in some situations if other network radios are the same).

CR205 Settings and Program

1. Configure all CR205s with the same hopping sequence, radio address, and network address using the same settings as the RF400 has.

2. Configure each CR205 with a unique PakBus address (“Datalogger Address” in PakCom) starting with “1” and ending with “5.”

3. Configure CR205 Power Mode = RF_ON.

   Configure CR205 Radio Force On = 0.

4. In LoggerNet CRBasic key in the following program for the CR205s:

```
Public AirTemp_C
Public RF_Strength

DataTable (AirTemp,1,1000)
  DataInterval (0,1,min)
  Average (1,AirTemp_C,0)
EndTable

BeginProg
Scan(10,sec)
'Make Measurements
Therm109 (AirTemp_C,1,1,Ex1,1.0,0)

RF_Strength=(Status.rfsignallevel(1,1)) ‘  Indicates CR205’s relative
received signal strength (RSSI)

CallTable AirTemp
NextScan
EndProg
```

5. Save it as TEMP_PROBE.dld and compile it.

6. Using PakCom, send TEMP_PROBE to each CR205 in network.

7. Connect power supply and antenna to each CR205.

8. Add 109-L probe to each CR205 or add 10KΩ/22KΩ resistors to simulate 109-L per Quick Start section Step 9.

<u>CR510PB Settings and Program</u>

1. In Edlog key in the following program:

```
*Table 1 Program
  01: 5           Execution Interval (seconds)

1: Batt Voltage (P10)
  1:  1           Loc [ Battery    ]

2: Internal Temperature (P17)
  1:  2           Loc [ PanelTemp   ]

3: Data Table (P84)
  1:  0           Seconds into Interval
  2:  10          Seconds Interval
  3:  0           (0 = auto allocate, -x = redirect to inloc x)
  4:              Battery_PanelTempAVGs    Table Name

4: Average (P71)
  1:  1           Reps
  2:  1           Loc [ Battery    ]

5: Average (P71)
  1:  1           Reps
  2:  2           Loc [ PanelTemp   ]
```

```
6:  Sample (P70)
   1:  1            Reps
   2:  12           Loc [ CR205RF_Stren3 ]

7:  PakBus - Get/Send Locations (P190)
   1:  17           SDC 7                        ; Same as CR510PB's RF400 Active Interface
   2:  1            Address                      ; CR205's PakBus Address
   3:  26           Get Value
   4:  0000         Security
   5:  0000         Remote Loc/Coil/Register
   6:  1            Swath
   7:  3            Local Loc [ CR205Temp_Dat1 ]
   8:  4            Result Code Loc [ StatusOfGet1   ]

8:  Extended Parameters 4 Digit (P68)
   1:  65           Option                       ;A
   2:  105          Option                       ;i
   3:  114          Option                       ;r
   4:  84            Option                      ;T
   5:  101          Option                       ;e
   6:  109          Option                       ;m
   7:  112          Option                       ;p
   8:  95           Option                       ;_

9:  Extended Parameters 4 Digit (P68)
   1:  0000         Option                       ;C
   2:  0000         Option                       ;
   3:  0000         Option
   4:  0000         Option
   5:  0000         Option
   6:  0000         Option
   7:  0000         Option
   8:  0000         Option

10:  PakBus - Get/Send Locations (P190)
   1:  17           SDC 7
   2:  2            Address
   3:  26           Get Value
   4:  0000         Security
   5:  0000         Remote Loc/Coil/Register
   6:  1            Swath
   7:  6            Local Loc [ CR205Temp_Dat2 ]
   8:  7            Result Code Loc [ StatusOfGet2   ]

11:  Extended Parameters 4 Digit (P68)
   1:  65           Option                       ;A
   2:  105          Option                       ;i
   3:  114          Option                       ;r
   4:  84           Option                       ;T
   5:  101          Option                       ;e
   6:  109          Option                       ;m
   7:  112          Option                       ;p
   8:  95           Option                       ;_
```

```
12:  Extended Parameters 4 Digit (P68)
  1:   0000          Option                          ;C
  2:   0000          Option                          ;
  3:   0000          Option
  4:   0000          Option
  5:   0000          Option
  6:   0000          Option
  7:   0000          Option
  8:   0000           Option

13:  PakBus - Get/Send Locations (P190)
  1:   17            SDC 7
  2:   3             Address
  3:   26            Get Value
  4:   0000          Security
  5:   0000          Remote Loc/Coil/Register
  6:   1             Swath
  7:   9             Local Loc [ CR205Temp_Dat3 ]
  8:   10            Result Code Loc [ StatusOfGet3   ]

14:  Extended Parameters 4 Digit (P68)
  1:   65            Option                          ;A
  2:   105           Option                          ;i
  3:   114           Option                          ;r
  4:   84            Option                          ;T
  5:   101           Option                          ;e
  6:   109           Option                          ;m
  7:   112           Option                          ;p
  8:   95            Option                          ;_

15:  Extended Parameters 4 Digit (P68)
  1:   0000          Option                          ;C
  2:   0000          Option                          ;
  3:   0000          Option
  4:   0000          Option
  5:   0000          Option
  6:   0000          Option
  7:   0000          Option
  8:   0000          Option

16:  PakBus - Get/Send Locations (P190); Get CR205 RF Signal Strength
  1:   17            SDC 7
  2:   3             Address
  3:   26            Get Value
  4:   0000          Security
  5:   0000          Remote Loc/Coil/Register
  6:   1             Swath
  7:   12            Local Loc [ CR205RF_Stren3 ]
  8:   13            Result Code Loc [ StatusOfGet3a   ]
```

```
17:  Extended Parameters 4 Digit (P68)
   1:    82            Option                           ;R
   2:    70            Option                           ;F
   3:    95            Option                           ;_
   4:    83            Option                           ;S
   5:    116           Option                           ;t
   6:    114           Option                           ;r
   7:    101           Option                           ;e
   8:    110           Option                           ;n

18:  Extended Parameters 4 Digit (P68)
   1:    103           Option                           ;g
   2:    116           Option                           ;t
   3:    104           Option                           ;h
   4:    0000          Option                           ;
   5:    0000          Option
   6:    0000          Option
   7:    0000          Option
   8:    0000          Option

*Table 2 Program
   02:  0.0000         Execution Interval (seconds)

*Table 3 Subroutines

End Program
```

2. Repeat lines 12 – 14 for the remaining CR205s.

3. Program Input Locations something like this in InLocEd:

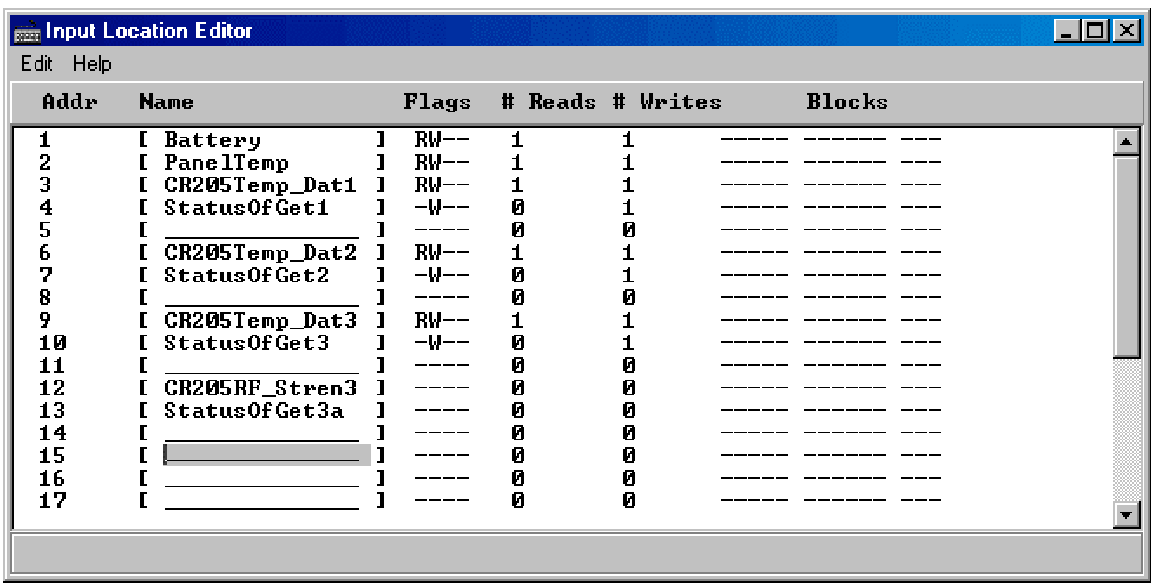


FIGURE E-1. P190 Input Locations Editor

4. Configure a CR510PB PakBus address of "100" like this inEdlog's Options\Pakbus Settings:

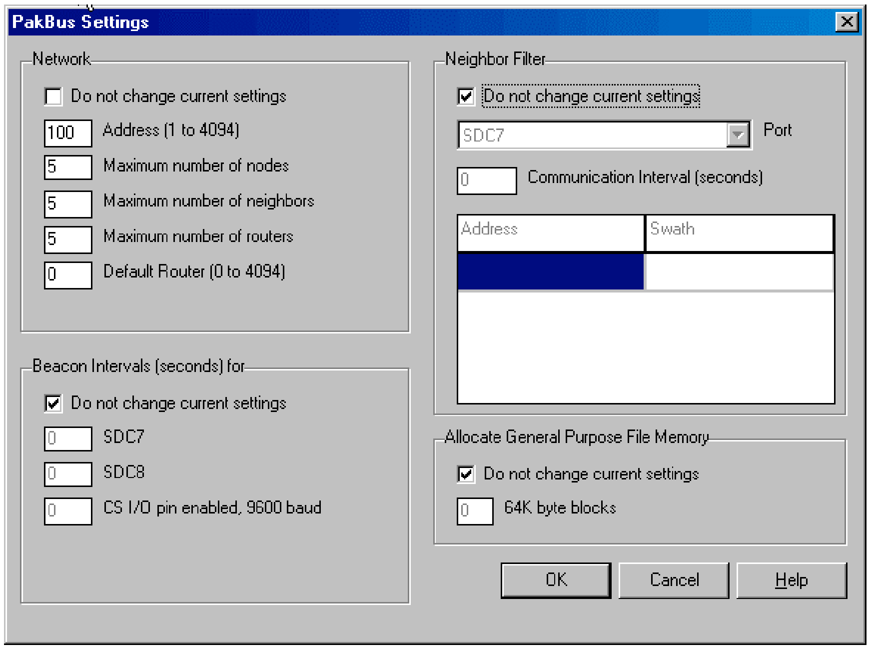


FIGURE E-2. P190 Edlog PakBus Settings

Configuring the maximum nodes, maximum neighbors, and maximum routers allows the CR510PB to function as a router in the network.

Notice that "Network" is the only setup group unchecked. There is no need of Neighbor Filter or Beacon setup because the P190 instruction creates a static route to each CR205 in the swath.

5. Save the program as P190TRANSFER.dld and compile it.

6. Send the program to the CR510PB.

7. Configure a Numeric Display in Connect Screen showing the CR510PB's Input Locations to monitor the remote CR205s' 109-Ls (or simulated 109-Ls). It can look something like this:

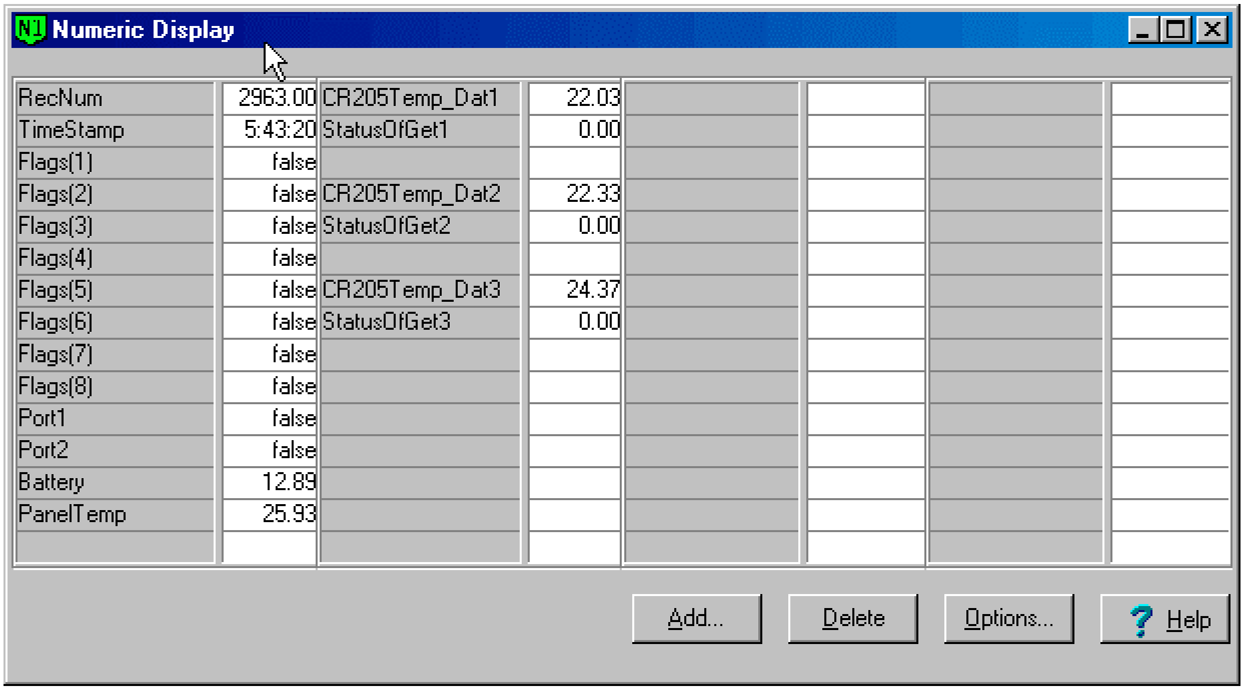


FIGURE E-3. P190 Numeric Display

If P190 communication is failing, the StatusOfGetx will increment every scan of the CR205. If zero, communication is successful.

Appendix F. CR510PB w/ P193 Master to CR205s Network

In this configuration LoggerNet does 15 minute scheduled collections of a remote CR510PB running a P193 to transfer variable values from 5 remote CR205s (may be other rf CR200 Series). This approach results in the lowest remote average current drain because the remotes originate the data transfer and their RF400 Series radios are off the rest of the time.

The P193 instruction has all the CR205s' addresses and does device discovery. A single P193 can communicate with many remotes having sequential PakBus addresses.

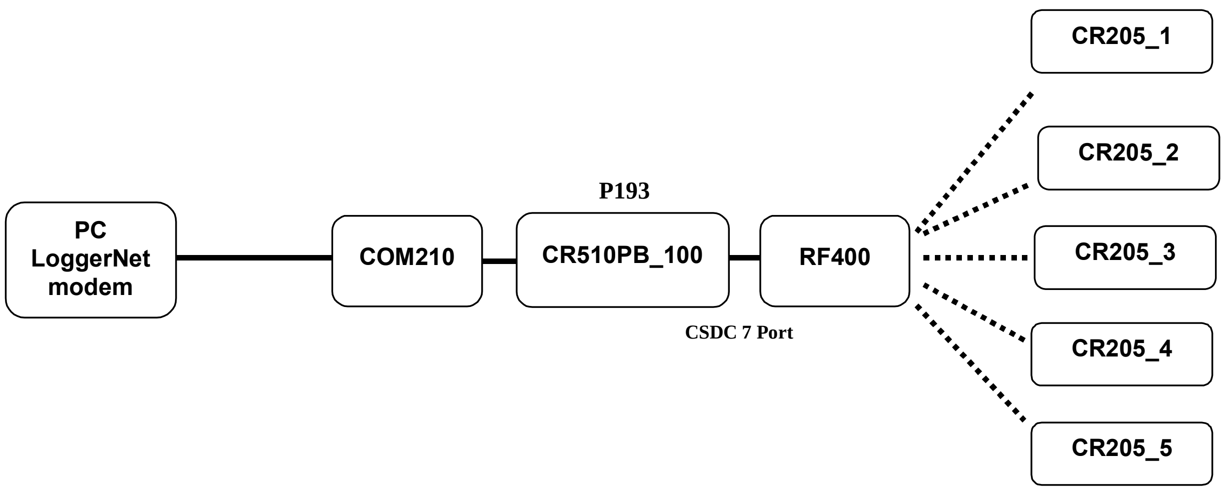
There is no beaconing in the described configuration. LoggerNet needs no beacon as it can discover CR510PB using the static route created from its device map. When LoggerNet is first run, it will be unaware of the CR510PB, but at the first connection they will exchange hellos and neighbor lists and become neighbors. LoggerNet can thereafter communicate with any node in the network. LoggerNet will keep the CR510PB as a neighbor until LoggerNet is closed or its router reset.

The example programs that follow demonstrate the ability of the P193 instruction to send data to a remote and receive data back from the remote.

NOTE In order for LoggerNet to send a program to a CR205 in the network, it may be necessary to temporarily set the CR510PB_100's P193 execution interval (program execution interval) to 3 minutes or more during the program transfer so that CR205 time-slot transmissions do not prevent the program transfer.

Be sure to collect CR510PB_100 data beforehand.

Hardware Setup



LoggerNet Device Map

ComPort_1
  PhoneBase
    PhoneRemote
      PakBusPort
        CR510PB (PakBus Address = 100)
          CR200_1 (PakBus Address = 1)
          CR200_2 (PakBus Address = 2)
          CR200_3 (PakBus Address = 3)
          CR200_4 (PakBus Address = 4)
          CR200_5 (PakBus Address = 5)

After once dialing up and connecting in LoggerNet, PakBusGraph will display the CR510PB and its CR205 neighbors.

LoggerNet Schedule Tab

Collection Interval – 15 minutes
Primary Retry Intervals – 5 minutes
Number of Primary Retries – 2
Secondary Retry Interval – 60 minutes

RF400 Settings

In a PakBus rf network, all RF400 Series radios are set up the same, except possibly the ports (Active Interfaces).

1. Configure RF400 radio with the same radio address, network address, and hopping sequence as the CR205s. If there will be no other RF400s in the area, you can use default settings, otherwise, choose some other hopping sequence, network address, and radio address and configure all radios in the network alike.

2. Other Settings

   Active Interface = “CSDC 7” (Port 7 is the default. Port 8 could be used)

Standby Mode = "< 24 mA always on" (other modes work in some situations if other network radios are the same).

CR205 Settings and Program

1. Configure all CR205s with the same radio address, network address, and hopping sequence using the same settings as the RF400 has.

2. Configure each CR205 with a unique PakBus address ("Datalogger Address" in PakCom) starting with address "1" and ending with address "5."

3. Configure CR205 Power Mode = RFpinEn
   (this is the minimum average power mode).

   Configure CR205 Radio Force On = 0

4. In LoggerNet CRBasic key in the following program for the CR205s:

```
'CR200

Public TimeUntilTx  'store time-slot and clock from Host
Public HostResponse  '0=success, -1=rcvd permit denied, -2=rcvd
insuff resources

Public Host(3)
Alias Host(1) = HostBattery
Alias Host(2) = HostTemp
Alias Host(3) = Number

Dim Remote(3)
Alias Remote(1) = Batt_volt
Alias Remote(2) = Therm_C
Alias Remote(3) = Therm_F
Public Thermister_C
Public Thermister_F
Dim Vout

DataTable(Table1,1,-1)
  DataInterval(0,1,min)
  Sample(1,Batt_Volt)
  Sample(1,Therm_C)
  Sample(1,Therm_F)
EndTable

BeginProg
Scan(1,sec)   ' TimeUntilTx requires a scan interval of 1 second ***

Therm109 (Thermister_C,1,1,Ex1,1.0,0)

Therm_C = Thermister_C
Therm_F = Therm_C*9/5 +32

Battery(Batt_Volt)

TimeUntilTx = TimeUntilTransmit(1)   '1 = RF com port, 2 = RS-232
com port
If TimeUntilTx = 0 Then
```

```
SendGetData (HostResponse,host(),Remote(),1,100,100,0)   'Host
response of 0 = success
Endif

CallTable(Table1)

NextScan
EndProg
```

5. Save the program as WIRELESS.CR2 and compile it.

6. Using PakCom (or LoggerNet), send WIRELESS.bin to each CR205 in network.

7. Connect power supply and antenna to each CR205.

CR510PB Settings and Program

1. In Edlog key in the following program:

```
*Table 1 Program
  01:  20             Execution Interval (seconds)

1:  Batt Voltage (P10)
  1:   1              Loc [ Battery      ]          ;  Fill Input Locs 1 to 3

2:  Internal Temperature (P17)
  1:   2              Loc [ Panel_Temp    ]

3:  Z=F x 10^n (P30)
  1:   123            F
  2:   0              n, Exponent of 10
  3:   3              Z Loc [ Number       ]

4:  Data Table (P84)
  1:   0              Seconds into Interval
  2:   60             Seconds Interval
  3:   0              (0 = auto allocate, -x = redirect to inloc x)
  4:   Transfer       Table Name

5:  Average (P71)
  1:   1              Reps
  2:   1              Loc [ Battery      ]

6:  Average (P71)
  1:   1              Reps
  2:   2              Loc [ Panel_Temp    ]
```

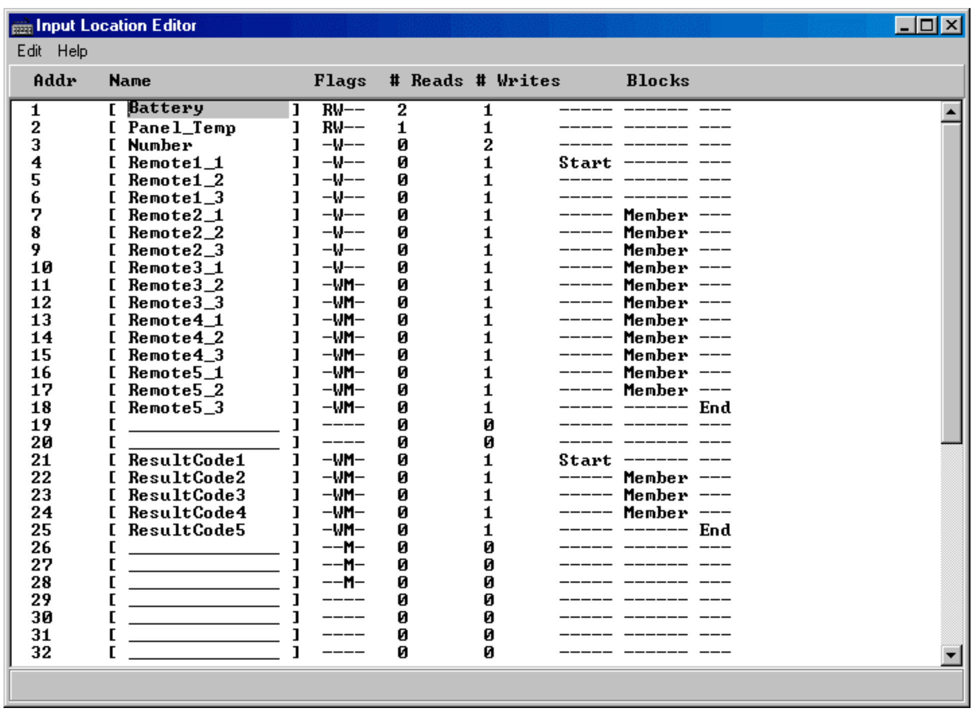
```
7:  PakBus - Wireless Network Master (P193)
   1:   5              Number of Remotes
   2:   1              First Remote Address           ; 1st CR205 "Datalogger Address"
   3:   0              Time Into Transmit Interval (sec)
   4:   0              Transmit Interval (sec, 0 = use execution interval) ;
                       Must = P193 execution interval
   5:   0              Transmit Delay Between Remotes (sec)
   6:   3              Swath to Receive               ; Values/remote
   7:   4              First Loc for Data Received [ Remote1_1     ]
   8:   3              Swath to Send
   9:   1              First Loc to Send [ Battery      ]
   10:  21             Result Code Loc [ ResultCode1    ]

*Table 2 Program
   02:  0.0000         Execution Interval (seconds)

*Table 3 Subroutines

End Program
```

Input Locations can be designed something like this:



Addr	Name	Flags	# Reads	# Writes	Blocks
1	[Battery]	RW--	2	1	----- ------ ---
2	[Panel_Temp]	RW--	1	1	----- ------ ---
3	[Number]	-W--	0	2	----- ------ ---
4	[Remote1_1]	-W--	0	1	Start ------ ---
5	[Remote1_2]	-W--	0	1	----- ------ ---
6	[Remote1_3]	-W--	0	1	----- ------ ---
7	[Remote2_1]	-W--	0	1	----- Member ---
8	[Remote2_2]	-W--	0	1	----- Member ---
9	[Remote2_3]	-W--	0	1	----- Member ---
10	[Remote3_1]	-W--	0	1	----- Member ---
11	[Remote3_2]	-WM-	0	1	----- Member ---
12	[Remote3_3]	-WM-	0	1	----- Member ---
13	[Remote4_1]	-WM-	0	1	----- Member ---
14	[Remote4_2]	-WM-	0	1	----- Member ---
15	[Remote4_3]	-WM-	0	1	----- Member ---
16	[Remote5_1]	-WM-	0	1	----- Member ---
17	[Remote5_2]	-WM-	0	1	----- Member ---
18	[Remote5_3]	-WM-	0	1	----- ------ End
19	[____________]	----	0	0	----- ------ ---
20	[____________]	----	0	0	----- ------ ---
21	[ResultCode1]	-WM-	0	1	Start ------ ---
22	[ResultCode2]	-WM-	0	1	----- Member ---
23	[ResultCode3]	-WM-	0	1	----- Member ---
24	[ResultCode4]	-WM-	0	1	----- Member ---
25	[ResultCode5]	-WM-	0	1	----- ------ End
26	[____________]	--M-	0	0	----- ------ ---
27	[____________]	--M-	0	0	----- ------ ---
28	[____________]	--M-	0	0	----- ------ ---
29	[____________]	----	0	0	----- ------ ---
30	[____________]	----	0	0	----- ------ ---
31	[____________]	----	0	0	----- ------ ---
32	[____________]	----	0	0	----- ------ ---

FIGURE F-1. P193 Input Locations Editor

To each CR205, connect a 109-L thermister temperature probe to SE1, ⏚, and EX1.

Or, you can simulate the 109-L using a 22KΩ resistor and a 10KΩ resistor (1% or 5% tolerance, 1/8 watt to 1/2 Watt). Connect the 22KΩ resistor

from SE1 to ⏚ and the 10KΩ resistor from EX1 to SE1. The result will be a simulated temperature reading of 22°C ± 2 degrees.

2. Configure a CR510PB PakBus address of "100" in Edlog's Options\Pakbus Settings:

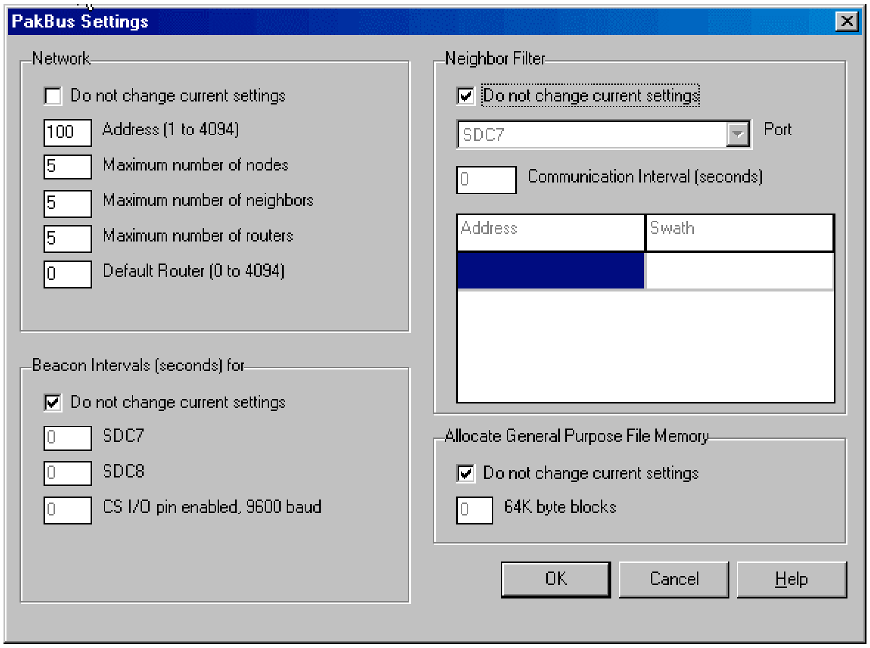


FIGURE F-2. P193 Edlog PakBus Settings

The maximum nodes, maximum neighbors, and maximum routers entries allow the CR510PB to function as a router in the network.

Notice that "Network" is the only setup group unchecked. There is no need of Neighbor Filter or Beacon setup because the P193 creates a static route to each CR205 in the swath.

3. Save and compile the program as P193TRANSFER.dld

4. Send the program to the CR510PB

5. Configure a Numeric Display in Connect Screen showing the CR510PB's Input Locations to monitor the remote CR205s' variables. You can click and drag to organize the display Input Locs to look something like this.

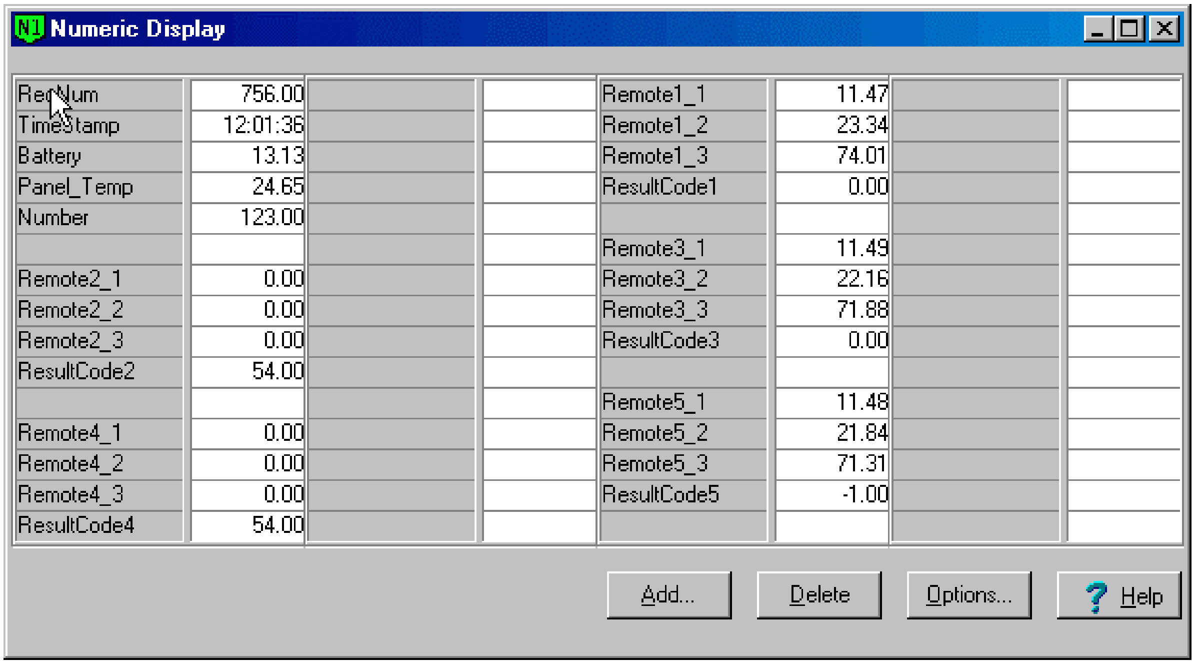


FIGURE F-3. P193 Numeric Display

Remote2 and Remote4 are missing in the above network, hence the “0.00” values and incrementing ResultCodes.

When the CR510PB receives a wireless message from a remote, the corresponding Result Code Location is set to -1. When Instruction 193 is executed, the Result Code Location is incremented by 1. Therefore, if communication is successful, the Result Code Location will be 0 after the execution of Instruction 193. If data transfer is unsuccessful, the Result Code Location for the remote that failed will be incremented, and will continue to increment with each failed attempt.

LoggerNet data collection of the CR510PB will slow somewhat due to the P193 traffic. For example, it took 5 minutes to collect 61K data points with a network similar to this Appendix’ network but with RF400 communications instead of COM210, 30 second P193 execution, and 7 remotes.

The RF400 in the above system can be configured with radio power mode (Standby Mode) of < 24 mA (radio always receiving) or < 4 mA (radio sleeping, awakening every ½ second). If you substitute a CR510PB with P196 for a CR205, its RF400 can also be configured for < 24 mA or < 4 ma. If the power budget is not tight, configuring < 24 mA on both ends improves response times slightly.

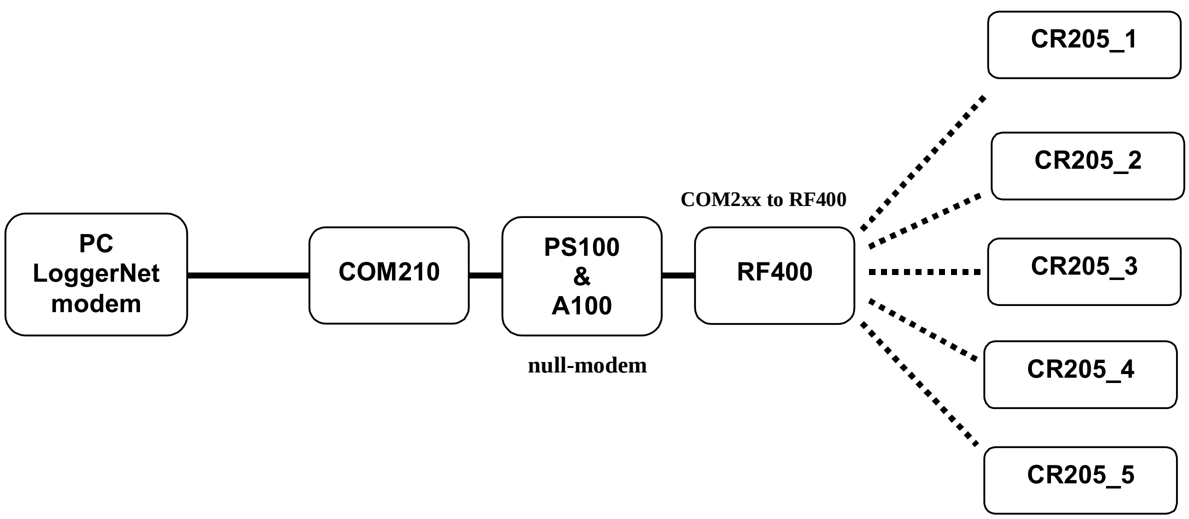
Appendix G. Phone to CR205s Network

In this configuration LoggerNet does direct 15 minute scheduled collections of 5 remote CR205s (may be other rf CR200 Series). This approach results in the greatest number of files collected from the remotes (as compared to the P190 or P193 approach).

There is no beaconing in this configuration. LoggerNet needs no beacon as it can discover CR205s using the static routes created by the device map. When LoggerNet is first run, it will be unaware of the CR205s, but at the first connection of each CR205 they will exchange hellos and become neighbors. LoggerNet will keep the CR205s as neighbors until LoggerNet is closed or its router is reset.

The COM210, PS100 with A100 and RF400 are not a PakBus node but merely a transparent connection for LoggerNet to talk, as though directly, to the CR205s.

Hardware Setup



LoggerNet Device Map

ComPort_1
  PhoneBase
    PhoneRemote
      PakBusPort
        CR200_1 (PakBus Address = 1)
        CR200_2 (PakBus Address = 2)
        CR200_3 (PakBus Address = 3)
        CR200_4 (PakBus Address = 4)
        CR200_5 (PakBus Address = 5)

After once dialing up and connecting in LoggerNet, PakBusGraph will display the CR205 neighbors.

LoggerNet Schedule Tab

Collection Interval – 15 minutes
Primary Retry Intervals – 5 minutes
Number of Primary Retries – 2
Secondary Retry Interval – 60 minutes

RF400 Settings

In a PakBus rf network, all RF400 Series radios are set up the same, except, possibly, the com ports.

1. Configure RF400 radio with the same network address, radio address, and hopping sequence as the CR205s. If there will be no other RF400s in the area, you can use default settings, otherwise, choose some other hopping sequence, network address, and radio address and configure all radios in the network alike.

2. Other Settings

   Active Interface = "COM2xx to RF400"

   Standby Mode = "<24 mA always on" (other modes work in some situations if other network radios are the same)

CR205s Settings and Program

1. Configure all CR205s with the same radio address, network address, and hopping sequence using the same settings as the RF400 has.

2. Configure each CR205 with a unique PakBus address ("Datalogger Address" in PakCom) starting with address "1" and ending with address "5."

3. Configure CR205s Power Mode = RF_ON

   Configure CR205s Radio Force On = 0

4. In LoggerNet CRBasic key in the following program for the CR205s:

```
'CR200

Public TimeUntilTx
Public HostResponse, Host(3) '0=success, -1=rcvd permit denied, -
2=rcvd insuff resources
Alias Host(1) = HostBattery 'store time-slot and clock from Host
Alias Host(2) = HostTemp

Dim Remote(3)
Alias Remote(1) = Batt_volt
Alias Remote(2) = Therm_C
Alias Remote(3) = Therm_F
Public Thermister_C
Public Thermister_F
Dim Vout
```

```
DataTable(Table1,1,-1)
  DataInterval(0,1,min)
  Sample(1,Batt_Volt)
  Sample(1,Therm_C)
  Sample(1,Therm_F)
EndTable

BeginProg
Scan(1,sec)

Therm109 (Thermister_C,1,1,Ex1,1.0,0)

Therm_C = Thermister_C
Therm_F = Therm_C*9/5 +32

Battery(Batt_Volt)

TimeUntilTx = TimeUntilTransmit(1)

If TimeUntilTx = 0 Then

SendGetData (HostResponse,host(),Remote(),1,100,100,0) 'Host
response of 0 = success

Endif

CallTable(Table1)

NextScan
EndProg
```

5. Save the program as WIRELESS.dld and compile it.

6. Using PakCom, send WIRELESS.dld to each CR205 in network.

7. Connect power supply and antenna to each CR205.

Data Viewing and Collection

You can set up Numeric Displays to view ‘temperature’ readings. Data collection can be scheduled.

Appendix H. Adding Direct or P190 Router to RF400 Network

If you have two locations that cannot be dependably rf linked using high gain antennas, tall tripods/towers or by optimizing antenna locations, sometimes adding a router is the best solution. The following shows how to add a router to an existing rf network using neighbor filter discovery.

For this router you will need:

1. CR510 with PakBus OS
2. RF400
3. PS12LA or equivalent 12 VDC power supply
4. 9591 18V/1.2A wall charger and 120 VAC source; or appropriate solar panel
5. SC12 cable (comes with RF400)
6. ENC 10/12 enclosure with appropriate mounting bracket, etc.
7. Antenna with appropriate antenna cable and surge suppressor
8. Tripod, tower or equivalent

Case 1 – Direct Collection of all dataloggers

Router Datalogger Configuration

- Configure CR510PB_1 as router in *D15 with PakBus Address (1), max nodes = 5, max neighbors = 2, max routers = 3 (includes LoggerNet).
- Configure CR510PB_1's *D19 neighbor filter listing the 'toward destination' device as a potential neighbor (LoggerNet counts as 1 router).
- Select datalogger port CSDC 7 (code 17) to hello potential neighbors.

Router RF400 Configuration

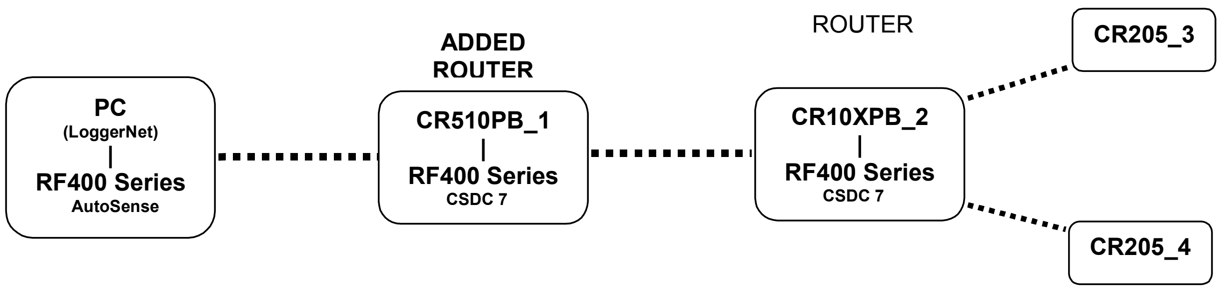
- Configure the RF400's Active Interface as CSDC (defaults to 7).
- Configure the RF400's Standby Mode to "< 24 mA always on" (other modes work in some situations if other network radios are the same)
- Configure CR510PB_1's RF400 net address, radio address, and hopping sequence the same as the network's.

Existing (router) Datalogger Configuration

- Edit the *D19 neighbor filter in CR10XPB_2 including the new router datalogger in its potential neighbor list. If you have two in-range routers using neighbor filters, in order for them to discover one another you must list each of them as a potential neighbor in the other's neighbor filter.

Antenna Height and Location

- Arrange the router antenna's height and/or position so that it has a good 'view' of the antennas of both the 'toward LoggerNet' and the 'toward destination' stations (height is everything in rf links).



Case 2 – P190 Transfers to/from CR205s

This amounts to keeping the P190 address normal but configuring the P190 datalogger's neighbor filter to discover the router and configuring the router's neighbor filter to discover the remote CR205. Refer to illustration below during setup.

Master Program

- Configure P190 to address destination (CR205_3) by entering "3" in Parameter 2.

Existing Master Datalogger Configuration

- Create neighbor filter in CR10XPB_1 listing added router CR510PB_2 as a potential neighbor.

- Make sure that CR205_3 is NOT in the neighbor filter potential neighbor list.

- Configure CR10XPB_1 as a router in *D15 with PakBus Address (1), max nodes = 5, max neighbors = 3, max routers = 3 (includes LoggerNet).

Added Router Datalogger Configuration

- Configure CR510PB_2 with neighbor filter listing CR205_3 as a potential neighbor.

- Configure CR510PB_2 neighbor filter listing CR10XPB_1 as a potential neighbor.

- In neighbor filter, select datalogger port CSDC 7 (code 17) to hello potential neighbor.

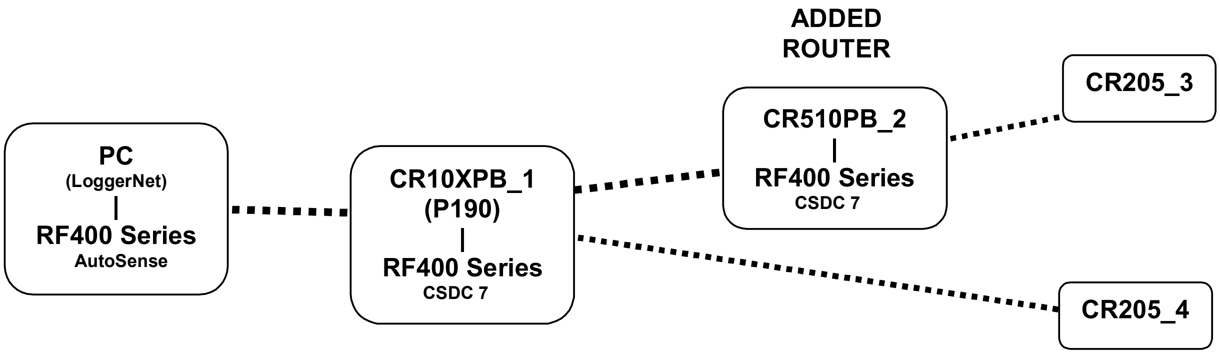
- Configure CR510_2 as a router in *D15 with PakBus Address (2), max nodes = 5, max neighbors = 2, max routers = 3 (includes LoggerNet).

Router RF400 Configuration

- Configure the RF400s' Active Interfaces to CSDC 7 (defaults to 7).
- Configure the RF400's Standby Mode to the Standby Mode setting that CR10XPB_1's RF400 and the CR205s use.
- Configure CR510PB_2's RF400 net address, radio address, and hopping sequence the same as the network's.

Antenna Height and Location

- Arrange the router antenna's height and/or position so that it has a good 'view' of the antennas of both the 'toward LoggerNet' and the 'toward destination' stations (height is everything in rf links).



NOTE It is possible to similarly add a router to a P193 path to remote.

Appendix P. PakBusGraph

PakBusGraph is a LoggerNet server client that allows a graphic view of a PakBus network from LoggerNet's perspective. This is a way to view LoggerNet's routing table. PakBusGraph also allows settings to be changed in nodes across the network, and it provides functions such as LoggerNet router reset and broadcast reset.

NOTE PakBusGraph is prototype software. If you own LoggerNet and desire a copy of PakBusGraph, please contact an application engineer for assistance.

PakBusGraph

PakBusGraph is a client of the LoggerNet server. It enables the user to see the routing table of the server. PakBusGraph can list network PakBus devices in the left panel, graphic representations of PakBus devices with links in the middle panel, and Log Messages resulting from changes in a PakBus device's neighbor list in the right panel.

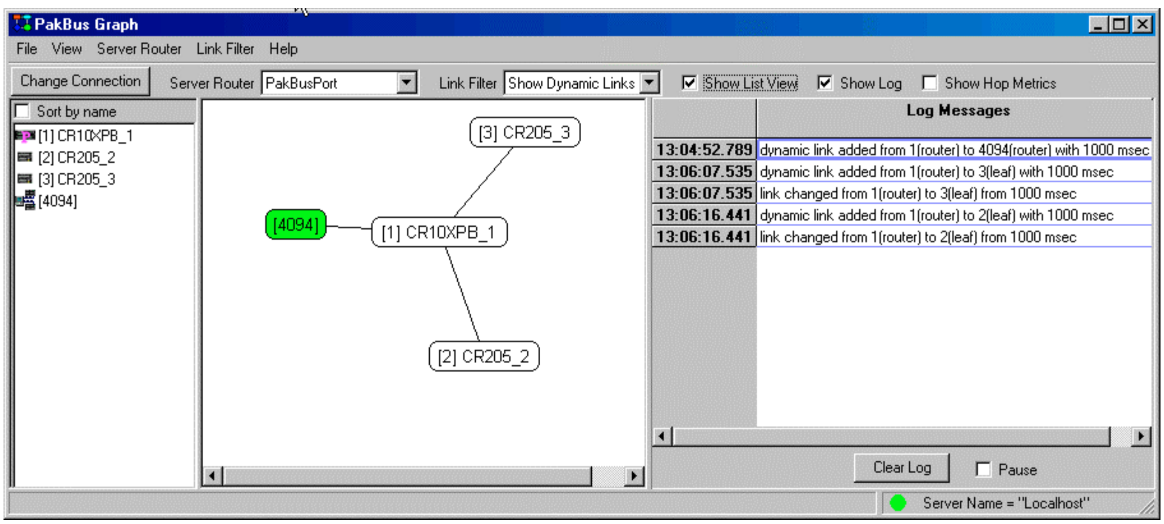


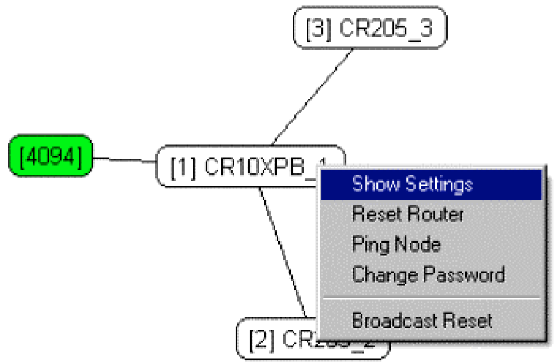
FIGURE P-1. PakBusGraph

You can choose which links you want to display: dynamic links (discovered), static links (as set up in LoggerNet), or all links. You can choose whether or not to show: the left panel List View, the Log Messages, and the Hop Metrics.

If a node is disabled in LoggerNet Setup, the PakBus Address will be displayed but not the station name.

View/Change Settings

Right-mouse click on a node and you will see, depending on the device, something like this:



In this example, a CR10XPB PakBus node was selected.

Show Settings

Displays:

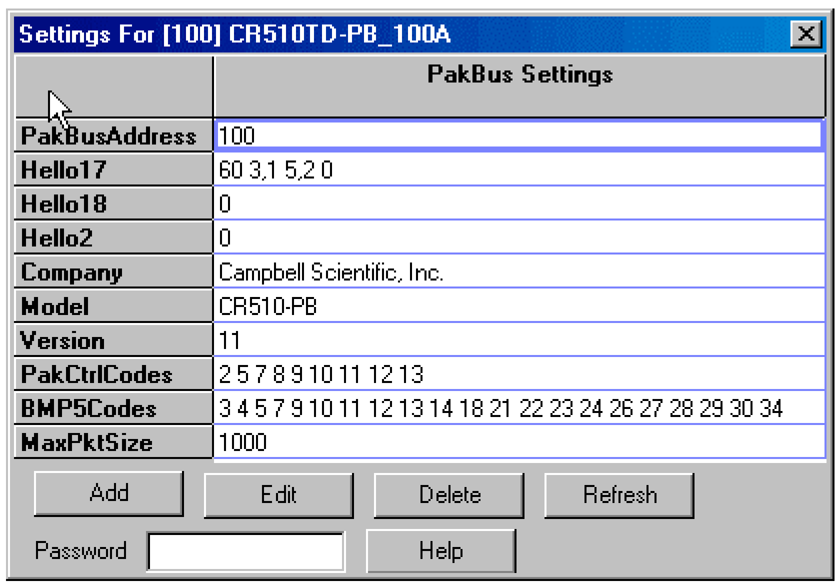


FIGURE P-2. Show Settings

PakBusAddress – Datalogger's own PakBus Address = 100

Hello17 (CSDC 7 Port) – Active with *D19 configured for 60 second communications verification interval; Potential Neighbors are PakBus device address 3 (swath of 1), and PakBus devices addresses 5 and 6 (5, swath of 2).

Hello18 (CSDC 8 Port) – not selected

Hello2 (M.E. Port) – not selected

Version – 11 is the version of OS in the datalogger if the device is a datalogger

PakCtrlCodes – shows the low level packet types supported

BMP5 Codes – shows the application level codes supported

Max Packet Size – advertises the max packet size this device can receive

Reset Router

Allows you to reset the Routing Table and Neighbor List of the selected PakBus Device so that it must re-learn its neighbors (by beacons, hello messages, P190, P191, P192, P196, P198, P224 or static routes).

Ping Node

This mode allows you to test if and how well LoggerNet is able to communicate with a PakBus device by sending/receiving packets of user variable size and frequency to/from the device. The right-hand Log Message panel shows the results of the pinging including a summary at the end of the number of packets sent, number received, and number lost. It includes the device response times.

Change Password

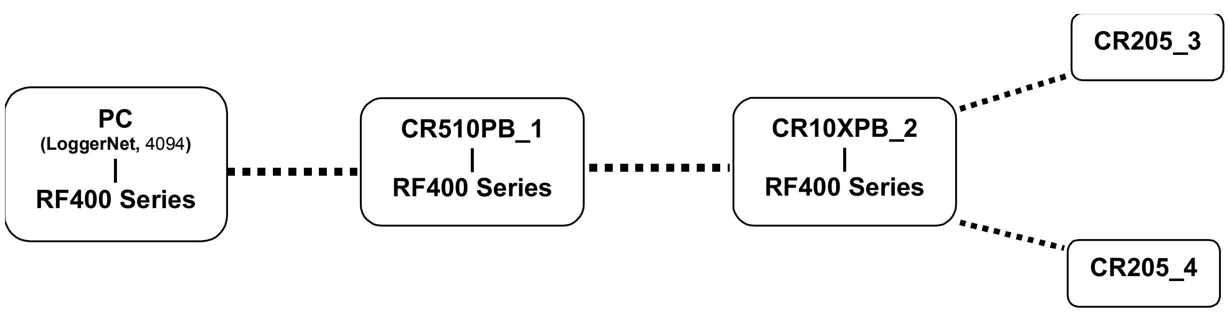
Allows you to change CR1000 PakBus Security.

Broadcast Reset

When you click on this it resets the Neighbor List for this PakBus device and tells all the neighbors of this PakBus device to reset their Neighbor Lists.

Hop Metrics

Hop Metrics also appear (if selected) in the interconnecting lines between PakBusGraph nodes showing the worst case response time (in milliseconds) for that link. For example, a PakBus Graph hop metric of 5000 is equivalent to a worst case response time of 5 seconds.



Usage

PakBusGraph can be used to set up or change PakBus settings from the PC or remotely using a laptop computer running LoggerNet with a different PakBus Address than the main PC (for example, laptop LoggerNet PakBus Computer ID = 4090).

Neighbor Filter:

If one of a neighbor filter's listed potential neighbors happens to fail, neighbor filter hello-ing will occur at random 1 to 15 second intervals until either the failed device is removed as a potential neighbor or communications is restored. Removal can be done from LoggerNet using PakBusGraph.

You can edit a node's neighbor filter by using PakBusGraph\Show Settings to edit the neighbor filter settings. Afterwards you should do a "Reset Router" to force the datalogger to soon re-learn its neighbors and links and pass the changes to routers in the network.

You can use PakBusGraph \ Settings to edit out Hello17, Hello18 or Hello2 potential neighbors and then select Reset Router to remove neighbors and links.

Neighbor List:

To manually reset a device's neighbor list, select the device in PakBusGraph, right-mouse click, and select Reset Router. If the device is configured to beacon, it will soon discover the neighbors again.

To manually reset a device's neighbor list, select the device in PakBusGraph, right-mouse click, and select Reset Router. If the device is configured to beacon, it will soon discover the neighbors again.

Routers:

To reset a LoggerNet, datalogger, or communications device router, you can right-click PakBusGraph and click on "Reset Router." This is a way to force a router to re-learn the network when troubleshooting.

PakBus Address:

A PakBus device's address can be changed (via *D15 or PakBusGraph). When this is done, a goodbye-message using the old address is broadcast out all ports which have a neighbor. This allows the network to update viable links.

Appendix Q. Log View

NOTE Log View is prototype software. If you own LoggerNet and desire a copy of Log View, please contact an application engineer for assistance.

LogView is a non-client software tool designed to interpret PakBus packets. Using the Low Level logs generated by the LoggerNet server (logs enabled in Status Monitor), you can view packet decoding to help you determine what is happening in the network. The logs are typically found in C:\Campbellsci\LoggerNet\Logs. The logs contain date/time stamps, so, if you notice the times of the packets of interest in the Low Level screen (to the nearest second), you will be able to find the corresponding packets and display them in LogView.

For example, you can open a log of type Low Level in LogView \ Open Log \ ComPort_1 and then go to View \ Add PakBus Filter to split the screen and click "Next" to decode the first line in the top of the upper screen to see lines of "bd" packets decoded into Message Type, Source address, Destination address, etc. Keep clicking the Next button to see the packets that follow.

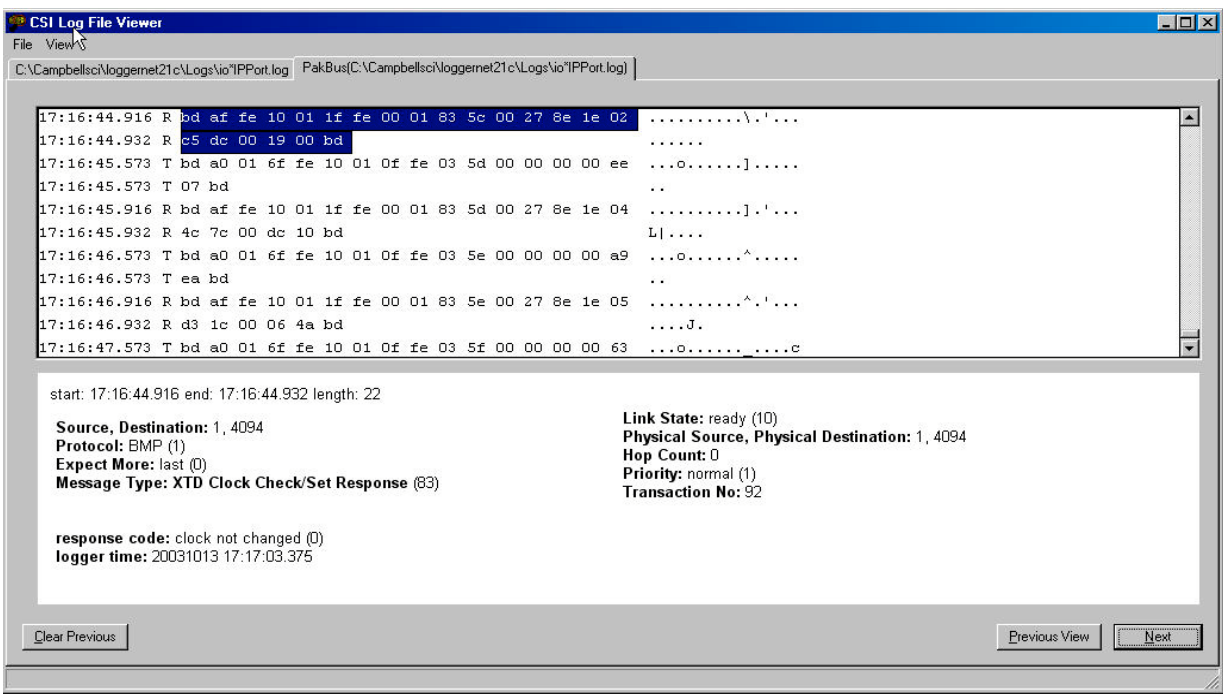


FIGURE Q-1. LogView Decodes Clock Check/Set Response (83) Packet

Source, Destination

The source is the device that originated the packet being examined.

The Destination is the device to which the packet ultimately goes.

Protocol

BMP refers to BMP5 packet protocol for PakBus. The "(1)"also refers to the BMP type of packet.

No protocol listed indicates the low-level (PakCtrl) protocol used for functions such as helloing, and neighbor list sharing.

Expect More

0 = last data in this direction

1 = more data to come in this direction

2 = neutral

3 = expect more in reverse direction

Message Type

The packet type. For example, XTD Clock Check/Set Response or Hello Command.

Is Router

Indicates whether the source PakBus device is configured as a router or not (PakCtrl).

Hop Metric

Indicates the worst case response time of the destination device (PakCtrl).

Beacon Interval

Indicates the beacon interval of packet source, if any (PakCtrl).

Link State

Indicates the link state of the source device as:

8 = Off-line

9 = Ring

a = Ready (10)

b = Finished (11)

c = Pause (12)

Physical Source, Physical Destination

Indicates the neighbor device who just sent the packet and the neighbor device who just received it.

Hop Count

Indicates how many routers the packet passes through from source to destination.

Priority

Indicates a packet's importance from 0 to 3. 3 indicates highest priority.

Transaction Number

Indicates a server serialized transaction number. This number follows a packet through to the destination and appears in response packets. For example a hello command packet might have transaction number 58. The hello response packet would also indicate a transaction number of 58. This helps the user track command and response.

There are various other indicators associated with Collect Data and other packets.

Campbell Scientific Companies

Campbell Scientific, Inc. (CSI)
815 West 1800 North
Logan, Utah 84321
UNITED STATES
www.campbellsci.com
info@campbellsci.com

Campbell Scientific Africa Pty. Ltd. (CSAf)
PO Box 2450
Somerset West 7129
SOUTH AFRICA
www.csafrica.co.za
sales@csafrica.co.za

Campbell Scientific Australia Pty. Ltd. (CSA)
PO Box 444
Thuringowa Central
QLD 4812 AUSTRALIA
www.campbellsci.com.au
info@campbellsci.com.au

Campbell Scientific do Brazil Ltda. (CSB)
Rua Luisa Crapsi Orsi, 15 Butantã
CEP: 005543-000 São Paulo SP BRAZIL
www.campbellsci.com.br
suporte@campbellsci.com.br

Campbell Scientific Canada Corp. (CSC)
11564 - 149th Street NW
Edmonton, Alberta T5M 1W7
CANADA
www.campbellsci.ca
dataloggers@campbellsci.ca

Campbell Scientific Ltd. (CSL)
Campbell Park
80 Hathern Road
Shepshed, Loughborough LE12 9GX
UNITED KINGDOM
www.campbellsci.co.uk
sales@campbellsci.co.uk

Campbell Scientific Ltd. (France)
Miniparc du Verger - Bat. H
1, rue de Terre Neuve - Les Ulis
91967 COURTABOEUF CEDEX
FRANCE
www.campbellsci.fr
campbell.scientific@wanadoo.fr

Campbell Scientific Spain, S. L.
Psg. Font 14, local 8
08013 Barcelona
SPAIN
www.campbellsci.es
info@campbellsci.es

Please visit www.campbellsci.com to obtain contact information for your local US or International representative.